

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

#2  
LDD  
2-90

10925 U.S. PTO  
09/695195



Applicant: Satoru FUJITA, et al.

Title: DISTRIBUTED APPLICATION CONTROL SYSTEM, CONTROL METHOD  
AND A PROGRAM

Appl. No.: Unassigned

Filing Date: October 25, 2000

Examiner: Unassigned

Art Unit: Unassigned

**CLAIM FOR CONVENTION PRIORITY**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application:

- Japanese Patent Application No. 11-304681 filed October 26, 1999.

Respectfully submitted,

Date October 25, 2000

By

Reg no 41514

FOLEY & LARDNER  
Washington Harbour  
3000 K Street, N.W., Suite 500  
Washington, D.C. 20007-5109  
Telephone: (202) 672-5407  
Facsimile: (202) 672-5399

David A. Blumenthal  
Attorney for Applicant  
Registration No. 26,257

FUJITA  
29471/1

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

US  
J-925 U.S. PTO  
09/695195

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
in this Office.

願 年 月 日  
Date of Application:

1999年10月26日

願 番 号  
Application Number:

平成11年特許願第304681号

願 人  
Applicant(s):

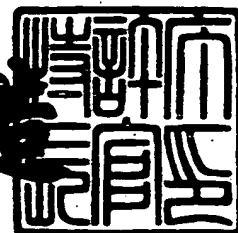
日本電気株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年 7月14日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



【書類名】 特許願

【整理番号】 33509610

【提出日】 平成11年10月26日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/44

G06F 13/00

F06F 15/16

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 藤田 悟

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 小山 和也

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100080816

【弁理士】

【氏名又は名称】 加藤 朝道

【電話番号】 045-476-1131

【手数料の表示】

【予納台帳番号】 030362

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9304371

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 分散アプリケーション制御システム及び制御方法並びにプログラムを記録した記録媒体

【特許請求の範囲】

【請求項 1】

複数の計算機がネットワークを介して相互に接続されてなるシステムにおいて

前記各計算機が具備するエージェントプラットフォームが、

分散アプリケーションを制御するためのスクリプト言語の入力を受け、前記スクリプト言語を解釈して実行するシェルエージェントと、

前記計算機にローカルなファイルシステムの情報を提供するローカルサービスエージェントと、

アプリケーションを制御するアプリケーションエージェントと、

を有し、前記シェルエージェント、ローカルサービスエージェント、及び、前記アプリケーションエージェントにそれぞれ実行の場を提供するエージェントベースと、

エージェントを他の計算機上のエージェントベースに移動する機能を提供するエージェント移動手段と、

エージェントが自計算機又は他の計算機の他のエージェントと通信を行うための機能を提供するリモートコール手段と、

アプリケーションエージェントを生成するエージェント生成手段と、

を備え、

前記スクリプト言語の入力に対して、前記各計算機に分散したアプリケーションの実行を制御する、ことを特徴とする分散アプリケーション制御システム。

【請求項 2】

前記シェルエージェントが、スクリプト言語を実行する上での現在のディレクトリを管理する現在ディレクトリと、

前記スクリプト言語を構文解析する構文解析手段と、コマンドを解析して実行するコマンド解析・実行手段とを備えたシェルインタプリタと、

を含み、

前記現在ディレクトリが、リモート計算機上のエージェントベースを指し示す時に、前記シェルエージェント自身を、前記エージェント移動手段によって、前記リモート計算機に移動して、アプリケーションエージェントを実行する、

ことを特徴とする請求項 1 記載の分散アプリケーション制御システム。

【請求項 3】

前記シェルエージェントが、エージェントのリポジトリパステーブルをさらに備え、

アプリケーションエージェントを、リポジトリパス上から検索して実行する、ことを特徴とする請求項 2 記載の分散アプリケーション制御システム。

【請求項 4】

前記シェルエージェントが、エージェントの状態を管理する状態テーブルをさらに備え、

前記状態テーブルが、エージェントの並列実行カウントと、端末実行フラグと、を備え、

前記現在ディレクトリの現在ベースが、リモート計算機である時に、エージェントの状態が並列実行状態ではなく、且つ、端末実行モードでもない場合に、シェルエージェントを、前記リモート計算機に移動させる、

ことを特徴とする請求項 2 又は 3 記載の分散アプリケーション制御システム。

【請求項 5】

前記シェルエージェントによるアプリケーションエージェント生成のために、前記各計算機にローカルな資源へのアクセスを補助するローカルサービスエージェントを呼び出す手段を備え、

アプリケーションエージェントを生成する時に、前記シェルエージェントが直接、前記エージェント生成手段を呼び出すのではなく、前記ローカルサービスエージェントを介してアプリケーションエージェントを生成する、

ことを特徴とする請求項 2 乃至 4 のいずれかに記載の分散アプリケーション制御システム。

【請求項 6】

前記シェルエージェントが、エージェント参照テーブルをさらに備え、

前記エージェント参照テーブル中に格納された外部に起動中のアプリケーションエージェントに対して、移動や命令実行の依頼を行う、

ことを特徴とする請求項 2 乃至 5 のいずれかに記載の分散アプリケーション制御システム。

【請求項 7】

前記アプリケーションエージェントが、前記計算機に依存したアプリケーションの起動手段を備え、

前記シェルエージェントの指示で、前記アプリケーションを生成して管理する、ことを特徴とする請求項 1、2、3、6 のいずれかに記載の分散アプリケーション制御システム。

【請求項 8】

前記アプリケーションエージェントが、前記計算機に備えられた拡張子アプリケーション対応テーブルを参照して、ファイルの拡張子の種類によって起動するアプリケーションを選択する手段を備え、

前記シェルエージェントの指示で、前記選択されたアプリケーションを生成して管理する、ことを特徴とする請求項 1、2、3、5、6、7 のいずれかに記載の分散アプリケーション制御システム。

【請求項 9】

前記エージェントプラットフォームが、スレッド生成手段をさらに備え、

入力されたスクリプト言語に、並列実行が指定されている場合、前記スクリプト言語を実行するのに必要な分のスレッドを生成して、アプリケーションの並列実行を制御する、ことを特徴とする請求項 1 記載の分散アプリケーション制御システム。

【請求項 10】

前記エージェントプラットフォームが、生成されたスレッドによるリモート計算機中のアプリケーションエージェントを生成する時に、前記リモート計算機中のエージェントベース中にサブシェルエージェントを有し、

前記サブシェルエージェントが、アプリケーションエージェント生成を代行す

る、ことを特徴とする請求項 9 記載の分散アプリケーション制御システム。

【請求項 11】

複数の計算機がネットワークを介して相互に接続されるシステムにおける分散アプリケーション制御方法であって、

前記各計算機が、それぞれのエージェントプラットフォームにおいて、シェルエージェントが、分散アプリケーションを制御するために入力されたスクリプト言語を解釈して実行し、

ローカルサービスエージェントが、計算機にローカルなファイルシステムに関する情報を提供し、

アプリケーションエージェントがアプリケーションを制御し、

前記シェルエージェントと、前記ローカルサービスエージェントと、前記アプリケーションエージェントとに対して、エージェントベースがそれぞれの実行の場を提供し、

エージェントが他の計算機の他のエージェントベースに、エージェント移動機能を介して移動自在とされ、

エージェントが自計算機又は他の計算機中の他のエージェントとリモートコール機能を介して通信を行い、

前記シェルエージェントは、入力されたスクリプト言語を解釈して、エージェント生成機能を介してアプリケーションエージェントの生成を制御し、

スクリプト言語の入力に対して、前記各計算機に分散したアプリケーションの実行を制御する、ことを特徴とする分散アプリケーションの制御方法。

【請求項 12】

前記シェルエージェントが、スクリプト言語を実行する上での現在のディレクトリを管理するとともに、入力されたスクリプト言語の構文解析と、コマンドの解析と実行を行い、

現在のディレクトリがリモート計算機上のエージェントベースを指し示す時にシェルエージェント自身を、前記リモート計算機に移動してアプリケーションエージェントを実行する、ことを特徴とする請求項 11 記載の分散アプリケーションの制御方法。



【請求項 13】

前記シェルエージェントが、さらに、エージェントのリポジトリパステーブルを参照して、アプリケーションエージェントを、リポジトリパス上から検索して実行する、ことを特徴とする請求項 11 又は 12 記載の分散アプリケーションの制御方法。

【請求項 14】

前記シェルエージェントが、エージェントの状態を管理する状態テーブルのエージェントの並列実行カウントと端末実行フラグの情報を用い、現在ディレクトリの現在ベースが、リモート計算機である時に、エージェントの状態が並列実行状態でなく、端末実行モードでもないときに、シェルエージェントをリモート計算機に移動させる、ことを特徴とする請求項 11 乃至 13 のいずれかに記載の分散アプリケーションの制御方法。

【請求項 15】

前記シェルエージェントが、アプリケーションエージェントの生成にあたり、前記シェルエージェントが直接に前記エージェント生成機能と呼び出すのではなく、各計算機にローカルな資源へのアクセスを補助するローカルサービスエージェントを介して、前記アプリケーションエージェントを生成する、ことを特徴とする請求項 11 記載の分散アプリケーションの制御方法。

【請求項 16】

前記シェルエージェントが、エージェント参照テーブルを有し、前記エージェント参照テーブル中に格納されている、外部に起動中のアプリケーションエージェントに対して、移動や命令実行の依頼を行う、ことを特徴とする請求項 11 乃至 15 のいずれかに記載の分散アプリケーションの制御方法。

【請求項 17】

前記アプリケーションエージェントが、計算機に依存したアプリケーションの起動を行う機能を備え、前記シェルエージェントの指示で前記アプリケーションを生成して管理する、ことを特徴とする請求項 11、12、13、16 のいずれかに記載の分散アプリケーションの制御方法。

【請求項 18】

前記アプリケーションエージェントが、前記各計算機に備えられた拡張子アプリケーション対応テーブルを用いて、ファイルの拡張子の種類によって起動するアプリケーションを選択し、前記シェルエージェントの指示で前記で選択されたアプリケーションを生成して管理する、ことを特徴とする請求項 11、12、13、15、16、17 のいずれかに記載の分散アプリケーションの制御方法。

【請求項 19】

前記エージェントプラットフォームが、スレッド生成機能を有し、入力されたスクリプト言語に並列実行の記述がある時に、並列実行に必要な分のスレッドを生成して、アプリケーションの並列実行を制御する、ことを特徴とする請求項 11 記載の分散アプリケーションの制御方法。

【請求項 20】

前記エージェントプラットフォームが、生成されたスレッドによるリモート計算機中のアプリケーションエージェントを生成する時に、リモート計算機中のエージェントベース中にサブシェルエージェントを生成し、前記サブシェルエージェントがアプリケーションエージェント生成を代行する、ことを特徴とする請求項 11 記載の分散アプリケーションの制御方法。

【請求項 21】

複数の計算機がネットワークを介して相互に接続され、

前記各計算機のエージェントプラットフォームが、分散アプリケーションを制御するためのスクリプト言語の入力を受けて、それを解釈し、実行するシェルエージェントと、

前記計算機にローカルなファイルシステム等の情報を提供するローカルサービスエージェントと、

アプリケーションを制御するためのアプリケーションエージェントと、

前記シェルエージェントと前記ローカルサービスエージェントと前記アプリケーションエージェントに実行の場を提供するエージェントベースと、

エージェントが他の計算機の他のエージェントベースに移動する機能を提供するエージェント移動手段と、

エージェントが、自計算機又は他の計算機の他のエージェントと通信を行うた

めの機能を提供するリモートコール手段と、

アプリケーションエージェントを生成するための機能を提供するエージェント生成手段と、を備え、スクリプト言語の入力に対して前記各計算機に分散したアプリケーションの実行を制御する分散アプリケーション制御システムにおいて、

前記シェルエージェント、前記ローカルサービスエージェント、前記アプリケーションエージェント、前記エージェントベース、前記エージェント移動手段、前記リモートコール手段、及び前記エージェント生成手段を、前記計算機で実行させるためのプログラムを記録した記録媒体。

【請求項 22】

複数の計算機がネットワークを介して相互に接続され、

前記各計算機のエージェントプラットフォームが、分散アプリケーションを制御するためのスクリプト言語の入力を受けて、それを解釈し、実行するシェルエージェントと、

前記計算機にローカルなファイルシステム等の情報を提供するローカルサービスエージェントと、

アプリケーションを制御するためのアプリケーションエージェントと、

前記シェルエージェントと前記ローカルサービスエージェントと前記アプリケーションエージェントに実行の場を提供するエージェントベースと、

エージェントが他の計算機の他のエージェントベースに移動する機能を提供するエージェント移動手段と、

エージェントが、自計算機又は他の計算機の他のエージェントと通信を行うための機能を提供するリモートコール手段と、

アプリケーションエージェントを生成するための機能を提供するエージェント生成手段と、を備え、スクリプト言語の入力に対して、前記計算機に分散したアプリケーションの実行を制御する分散アプリケーション制御システムにおいて、

前記シェルエージェント、前記ローカルサービスエージェント、前記アプリケーションエージェント、前記エージェントベース、前記エージェント移動手段、前記リモートコール手段、及び、前記エージェント生成手段を、前記計算機で実行させるためのプログラムを担持する媒体。

【請求項 2 3】

1 又は複数の他の計算機とネットワークを介して接続される計算機が、  
エージェントプラットフォームと、前記計算機にローカルなファイルシステム  
と、アプリケーションと、エージェントリポジトリとを備え、

前記エージェントプラットフォームは、

前記計算機への入力を受け取る入力部と、

前記計算機からの出力を行う出力部と、

入力されたスクリプト言語を解釈して実行し、アプリケーションエージェント  
の生成と通信を行うシェルエージェントと、

前記計算機に固有の情報や機能を提供するエージェントであり前記ローカルフ  
ァイルシステムを管理してシェルエージェントに対して情報の提供を行うローカ  
ルサービスエージェントと、

前記シェルエージェントからの要求で所定のタスクを実行するアプリケーショ  
ンエージェントと、

前記各エージェントに対して実行中のエージェントプログラムを格納する領域  
を提供するエージェントベースと、

前記エージェントベース中のエージェントが、他のエージェントプラットフォ  
ームに移動する機能を提供するエージェント移動部と、

エージェントベース中のエージェントが、マルチスレッドで動作する時に、新  
しいスレッドを生成するための機能を提供するスレッド生成部と、

前記エージェントベース中のエージェントが、同じエージェントプラットフォ  
ーム中の別のエージェント、あるいは、他のエージェントプラットフォームのエ  
ージェントに対して、メソッド呼び出しを行う機能を提供するリモートコール部  
と、

新しくエージェントを前記エージェントベースの中に生成して実行する時に呼  
び出され、前記エージェントリポジトリを参照して、エージェントプログラムを  
検索し、検索結果に基づいて、前記エージェントベースの中に新しくエージェン  
トを生成するエージェント生成部と、

を備え、

前記入力部を通して入力されたスクリプト言語を、前記シェルエージェントが解釈して、前記アプリケーションエージェントを起動し、前記アプリケーションエージェントが実際のアプリケーションを管理し、

前記シェルエージェント、及び、前記アプリケーションエージェントは、前記エージェント移動部を用いて、前記ネットワークを介して、他の計算機に移動自在とされるとともに、前記リモートコール部を介して、前記ネットワークを介して他の計算機中のエージェントと、通信可能とされている、ことを特徴とする計算機。

【請求項 24】

前記シェルエージェントのシェルインタプリタが、前記入力部からスクリプト言語の入力を受けこれを解釈し、並列実行文である場合、並列実行処理を行い、

内部コマンドとして処理できる場合には、内部コマンド処理を行い、前記内部コマンドでない場合には、外部コマンド処理を行うように制御する、ことを特徴とする請求項 23 記載の計算機。

【請求項 25】

前記シェルエージェントが、

前記入力部からスクリプト言語の入力を受け、入力されたスクリプト言語を解釈する構文解析部と、コマンドを実行するコマンド解析・実行部と、を具備するシェルインタプリタと

前記シェルエージェントの実行状態を管理するための状態テーブルと、

前記シェルエージェントの現在ディレクトリを記憶する現在ディレクトリ記憶部と、

を備え、

前記構文解析部は、キーワードを登録したキーワードテーブルを参照し、並列実行文のキーワードで始まる構文を持つ入力を受理した場合には、前記スレッド生成部に対して新しいスレッドの生成を依頼して、並列実行を行い、

前記コマンド解析・実行部は、入力されたコマンドが、前記シェルエージェント内部で処理可能な内部コマンドであるか否かを判定し、判定の結果、内部コマンドの場合、内部コマンド定義を記憶した内部コマンド定義記憶部からコマンド

の定義をロードして実行し、その際、前記内部コマンド定義記憶部からロードされた内部コマンドのうち、ディレクトリ移動のコマンドを実行するにあたり、前記シェルエージェント自身の移動を必要とする場合、前記エージェント移動部を呼び出し、

他のエージェントのメソッドを実行する場合には、前記リモートコール部を利用し、

前記コマンド解析・実行部によるコマンド実行の結果は、前記出力部に渡されて、前記計算機から出力され、

一方、外部コマンドの実行の場合、前記シェルエージェントに設けられた外部コマンド実行部にて、エージェントリポジトリより、外部コマンドを実行するエージェントプログラムを検索し、その際、前記エージェントリポジトリ内のどのパスを検索するか否かを決定するために、リポジトリパステーブルを参照し、前記リポジトリパステーブルに格納されたパスについて、順次、前記エージェントリポジトリを検索し、

前記外部コマンド実行部は、前記エージェントリポジトリを検索した結果、必要なエージェントが検索された場合、前記エージェント生成部に対して、エージェントの生成を依頼し、この依頼を受けた前記エージェント生成部が、アプリケーションエージェントを生成し、

さらに、前記外部コマンド実行部は、前記シェルエージェント以外のエージェントにメソッドコールを行う場合、前記リモートコール部を呼び出す、ことを特徴とする請求項 2 3 記載の計算機。

#### 【請求項 2 6】

前記ローカルサービスエージェントが、前記外部コマンド実行部から、外部コマンド名と、前記リポジトリパステーブルに格納されたパス情報とを受け取り、前記エージェントリポジトリの検索と、実際のエージェント生成部の呼び出しを行うことで、前記アプリケーションエージェントを生成する、ことを特徴とする請求項 2 5 記載の計算機。

#### 【請求項 2 7】

前記内部コマンドがディレクトリ移動を表すコマンドである場合、移動先ディ

レクトリを前記内部コマンドから抽出してディレクトリ構成要素に展開し、ベース名をディレクトリ格納テーブルの現在ベースに、ベース内のディレクトリを、ベース内ディレクトリにそれぞれ格納し、

前記状態テーブルに設けられている、並列実行の深さ情報を格納する並列実行カウンタの値が0であるか否かを検査し、前記並列実行カウンタが0以外の時は、シェルエージェントが並列実行していることを表しており、前記現在ディレクトリ記憶部の値を、前記ディレクトリ移動を表すコマンドで指定された移動先ディレクトリの値に更新し、シェルエージェントは計算機間で移動させず、

前記並列実行カウンタが0である場合には、シェルエージェントは単一スレッドで実行していることから、前記状態テーブルに設けられ、シェルエージェントが端末から入力待つシェルとして実行されている時は第1の値、バッチコマンドとして特定の端末に接続されないで実行されている時は第2の値に設定される端末モードフラグを参照し、前記端末モードフラグが第1の値のとき、前記ディレクトリ移動を表すコマンドによるシェルエージェントの計算機間移動は行わず、前記現在ディレクトリ記憶部の値の更新だけを行い、

前記端末モードフラグが第2の値の時は、バッチモードの実行であるため、前記現在ディレクトリ記憶部の値と、移動先ディレクトリとを比較し、前記現在ディレクトリ記憶部と移動先ディレクトリとが同一の計算機上にある場合には、前記現在ディレクトリ記憶部の値の更新だけを行い、前記現在ディレクトリ記憶部と移動先ディレクトリとが互いに同一の計算機にない場合には、前記シェルエージェントを、前記移動先ディレクトリの現在ベースが存在する計算機のエージェントベースに移動するように、制御する、ことを特徴とする請求項25記載の計算機。

#### 【請求項28】

前記シェルインタプリタの構文解析部において、並列実行文が検出され、並列実行処理を実行するにあたり、

前記状態テーブルの並列実行カウンタに1を加えてシェルエージェントが現在並列実行中であることを記録し、

前記並列実行文中のコマンド数分だけのスレッドを前記スレッド生成部によっ

て生成し、

スレッド変数に生成したスレッド数を記録設定し、

前記並列実行文の中のコマンドを各スレッドで並列に実行し、各スレッドの実行が終了すると、記録されたスレッド変数の値を 1 だけ減少させ、全てのスレッドが終了した時点で前記スレッド数が 0 となることで並列実行を終了し、

前記並列実行カウンタの値を 1 デクリメントして、並列実行処理全体の実行を終了する、ことを特徴とする請求項 2 3 記載の計算機。

#### 【請求項 2 9】

前記外部コマンド実行部が、前記現在ディレクトリ記憶部に格納されている現在ベースが、実際に、シェルエージェントを実行している実行中ベースと一致しているかを判断し、一致している場合には、実行中ベースに外部のアプリケーションエージェントを生成するため、まず、実行中ベースのエージェントリポジトリのエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断し、アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリからアプリケーションエージェントのプログラムをロードしてアプリケーションエージェントを実行するように、前記エージェント生成部に対して依頼し、

前記現在ディレクトリ記憶部に格納されている現在ベースとシェルエージェントを実行している実行中ベースとが一致していない場合には、前記リモートコール部を用いて、現在ベース中にサブシェルエージェントを作成し、

前記サブシェルエージェント中で現在ベースのエージェントリポジトリのエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するかを判断し、前記アプリケーションエージェントのプログラムがあれば、前記エージェントリポジトリからアプリケーションエージェントのプログラムをロードしてアプリケーションエージェントを実行するようにエージェント生成部に依頼する、ことを特徴とする請求項 2 5 記載の計算機。

#### 【請求項 3 0】

前記現在ディレクトリ記憶部に格納されている現在ベースが、実際にシェルエージェントを実行している実行中ベースと一致しているか否かを判断し、一致し



ている場合には、実行中ベースに外部のアプリケーションエージェントを生成するため、まず、実行中ベースの前記ローカルサービスエージェントに外部エージェントの実行依頼を行い、

前記ローカルサービスエージェントは、実行中ベースのエージェントリポジトリのエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断し、前記アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリからアプリケーションエージェントのプログラムをロードしてアプリケーションエージェントを実行するように、前記エージェント生成部に対して依頼し、

前記現在ディレクトリ記憶部に格納されている現在ベースと、実際にシェルエージェントを実行している実行中ベースとが一致していない場合には、前記リモートコール部を用いて現在ベース中の前記ローカルサービスエージェントに外部エージェントの実行依頼を行い、

前記ローカルサービスエージェントは、現在ベースのエージェントリポジトリのエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断し、アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリからアプリケーションエージェントのプログラムをロードしてアプリケーションエージェントを実行するように、エージェント生成部に依頼する、ことを特徴とする請求項 2 3 記載の計算機。

### 【請求項 3 1】

前記アプリケーションエージェントとして、汎用アプリケーションエージェントを備え、指定されたファイルの拡張子から、拡張子と該拡張子に対応するアプリケーションとの関係を記憶保持する拡張子アプリケーション対応テーブルを参照して検索し、前記ファイルの拡張子と一致した拡張子を持つアプリケーションを起動する、ことを特徴とする請求項 2 3 記載の計算機。

### 【発明の詳細な説明】

#### 【0 0 0 1】

#### 【発明の属する技術分野】

本発明は、情報処理システムに関し、特に、エージェントを用いた分散アプリ

ケーション制御システム、制御方法、及びプログラムを記録した記録媒体に関する。

【0002】

本発明は、後の説明でも明らかとされるように、複数の計算機に分散して存在するファイルを統一的なディレクトリに管理し、各ファイルの提供するサービスを実施するアプリケーションを管理するアプリケーションエージェントを、移動可能な制御用エージェントシステムが、そのファイルの存在する計算機に移動あるいは通信してスクリプトを実行することで、アプリケーション制御を行う分散アプリケーション制御システムに適用して好適とされる技術に関する。

【0003】

【従来の技術】

例えば、サンマイクロシステムズ社(Sun Microsystems)のNFS (Network File System) は、よく知られているように、ネットワーク接続された複数の計算機において、リモート計算機のファイルシステムを、計算機のローカルなファイルシステムであるかのように見せてディレクトリ管理し、当該計算機ではローカルなファイルシステムと同じように、リモート計算機のファイルにアクセスすることを可能としている。しかしながら、ファイルに対する処理は、ローカルな計算機によって行われ、リモート計算機上でファイルに対する処理を実行することはできない。

【0004】

また、例えば文献1 (「UNIXネットワークプログラミング、W.リチャード・スティーンズ著、プレントイスホール、1990」) 等に記載されているように、UNIXシステムに実装されているリモートシェルは、リモート計算機上でコマンドを実行するためのスクリプト言語である。このスクリプト言語では、実行する計算機を指定し、該指定された計算機上で、スクリプトで指定されたコマンドを実行する。ただし、一つのスクリプトは、一つの計算機上でだけで解釈される。

【0005】

複数の計算機上でコマンドを実行するには、それぞれが実行対象計算機を指定した複数のスクリプトを用意し、複数のスクリプトを逐次的に、複数の計算機の

それぞれに対して、リモートシェルとして呼び出すことで実現するしか、手だてがない。

【0006】

ところで、複数の計算機にまたがって、実行を継続する技術として、移動エージェントシステムが知られている。

【0007】

例えば文献2（特開平7-182174号公報）に記載されているように、米国ジェネラルマジック社のテレスクリプトの技術では、エージェントが「go」命令によって、他の計算機を移動する。エージェントが、他のエージェントと通信を行うためには、事前に相手のエージェントが存在する「プレース」という特殊な場所に移動する必要がある、逆に言えば、同じプレースに存在するエージェント間だけで通信を行うことができる。複数の計算機上を巡回してサービスを受けることができることが特徴的であるが、プレース管理と、サービス管理が別々であったこと、複数の計算機に対して、一つのエージェントが並列にアクセスすることができないことなどの制限があった。

【0008】

また文献3（「William Cockayne, Michael Zyda 著, Mobile Agent, Manning Publications Co., 1997」）に記載されている「Agent Tcl」、あるいは、別名「D'Agent」は、米国ダートマス大学で開発された移動エージェントである。この移動エージェントは、「Tcl」と呼ばれるシェルスクリプト言語を基にしており、これに、「agent\_jump」（エージェントジャンプ）命令による移動機能等が付加されている。「Tcl」は簡易シェルとして利用できるものであり、テレスクリプトのようなプログラム言語と比べて、利用者に簡単に利用できるシェルスクリプト言語である。

【0009】

しかしながら、「Agent Tcl」においても、エージェントの計算機の移動は、移動命令により行われており、計算機内の資源とは、別途ディレクトリで管理されており、計算機と、計算機内の資源とは、それぞれ、別のディレクトリで管理されており、移動命令と計算機資源の管理は独立に行われている。

【0 0 1 0】

また、一つのエージェントは、同時に一つの計算機にしか存在することができないため、複数の計算機の資源を並列的に扱うことはできない。

【0 0 1 1】

さらに、例えば文献 4（特開平 1 0 - 1 4 9 2 8 7 号公報）には、「goto」命令によりエージェントを移動できるスクリプト言語のアクションを述語として記述できる述語論理型の移動エージェントシステムが開示されている。

【0 0 1 2】

上記文献 4 に記載されるシステムにおいて、スクリプト言語は、エージェントの移動を行う記述を書くことができるが、計算機が提供する資源と合わせてディレクトリ管理することはできない。また、複数の計算機の資源に並列にアクセスできない点も、他の移動エージェント言語と同様である。

【0 0 1 3】

そして、例えば文献 5（特願平 1 1 - 0 4 7 0 1 5 号：本願出願時未公開）には、エージェントの移動命令を持ち、実行中のプログラムを他の計算機上に移動させることができるだけでなく、一つのエージェントが複数の計算機上に分散して、並列的に実行を行う機能を備え、一つのプログラムが、複数の計算機に分散して存在するサービスを統合して分散かつ同期的に管理し、ネットワークにまたがる一つのサービスとして提供することを可能とするシステムが記載されている。しかしながら、移動命令と計算機資源の管理が互いに独立に行われており、ネットワークに分散した計算機資源を、利用者に簡単に見せることはできない。

【0 0 1 4】

【発明が解決しようとする課題】

上記したように、上記文献等に記載されている技術は、下記記載の問題点を有している。

【0 0 1 5】

第 1 の問題点は、複数のリモート計算機とその計算機の提供する資源を、簡易なディレクトリ構造で管理して、ディレクトリを移動することで、資源のある場所にその資源を管理するエージェントを生成し、資源の制御を対話的に行うこと

ができる制御環境を提供していない、ということである。

【 0 0 1 6 】

その理由は、移動エージェントを用いていないシステムでは、計算機間を簡易に移動する機能を持ち合わせず、リモート計算機上に資源管理用のエージェントを作成することができないためである。

【 0 0 1 7 】

また移動エージェントを用いるシステムであっても、リモート計算機との通信は制限されていたり、あるいは、計算機間の移動と、計算機内資源のディレクトリ管理は、別個に管理されている、ためである。

【 0 0 1 8 】

第2の問題点は、従来の移動エージェントシステムにおいて、複数のリモート計算機の提供するサービスを、一つの制御用エージェントから、逐次的に制御する手段と、分散かつ同期して制御する手段の両者を具備することができない、ということである。

【 0 0 1 9 】

その理由は、上記文献等に記載されている移動エージェントシステムは、移動手段によって、逐次的に複数の計算機を巡回する機能を具備しているものの、一つのエージェントが複数の計算機に分散して存在し、複数のサービスを制御するという機能は具備していないためである。

【 0 0 2 0 】

また、移動機能と分散機能を持つエージェントシステムにおいても、その目的は、エージェントシステムの開発用プログラム言語の構築であって、逐次制御と分散制御を利用者が容易に組み合わせることを主眼においたスクリプト言語の開発という観点を持った開発が存在していないためである。

【 0 0 2 1 】

したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、利用者との対話的な環境によって、複数の計算機上の資源を管理するエージェントの生成、移動、終了を、スクリプト言語を用いて、簡単に制御できるシステム及び方法並びに記録媒体を提供することにある。

## 【 0 0 2 2 】

本発明の他の目的は、リモート計算機上のエージェントの制御を行うシステムに、移動機能と分散機能を持たせ、利用者には、計算機移動と資源管理を一つのディレクトリ構造に見せることで、計算機ネットワークに関して知識を持たない利用者でも、移動機能と分散機能の必要な統合サービスを実現するエージェントを、スクリプト言語によって簡易に記述できるシステム及び方法並びに記録媒体を提供することにある。これ以外の本発明の目的、特徴、利点等は以下の説明から、当業者には、直ちに明らかとされるであろう。

## 【 0 0 2 3 】

## 【課題を解決するための手段】

前記目的を達成する本発明に係るシステムは、分散アプリケーションを制御するためのスクリプト言語の入力を受けて前記スクリプト言語を解釈して実行するシェルエージェントと、計算機にローカルなファイルシステムの情報を提供するローカルサービスエージェントと、アプリケーションを直接制御するためのアプリケーションエージェントと、前記シェルエージェントと前記ローカルサービスエージェントと前記アプリケーションエージェントに実行の場を提供するエージェントベースと、エージェントを他の計算機上のエージェントベースに移動させるエージェント移動手段と、エージェントが自計算機又は他の計算機中の他のエージェントと通信を行うリモートコール手段と、アプリケーションエージェントを生成するエージェント生成手段と、前記エージェントベース、前記エージェント移動手段、前記リモートコール手段、前記エージェント生成手段を備えるエージェントプラットフォームと、を備えて構成され、前記スクリプト言語の入力に対して各計算機に分散したアプリケーションの実行を制御する。

## 【 0 0 2 4 】

本発明に係る分散アプリケーション制御方法は、シェルエージェントが、分散アプリケーションを制御するために入力されたスクリプト言語を解釈して実行し、ローカルサービスエージェントが、計算機にローカルなファイルシステムに関する情報を提供し、アプリケーションエージェントがアプリケーションを直接制御し、前記シェルエージェントと、前記ローカルサービスエージェントと、前記

アプリケーションエージェントとに対してそれぞれエージェントベースが実行の場を提供し、エージェントが別の計算機の別のエージェントベースに移動自在とされ、エージェントが自計算機又は他の計算機中の他のエージェントと通信を行い、前記シェルエージェントは入力されたスクリプト言語を解釈してアプリケーションエージェントの生成を制御し、エージェントプラットフォームが前記エージェント生成手段を管理し、スクリプト言語の入力に対して、前記各計算機に分散したアプリケーションの実行を制御する。

【0025】

#### 【発明の実施の形態】

本発明の実施の形態について以下に説明する。本発明の分散アプリケーション制御システムは、一実施の形態において、スクリプト言語を解釈するシェルエージェントが自らを移動させるか、あるいは、リモート計算機のエージェントプラットフォームと通信することによって複数の計算機に分散したアプリケーションを、一つのスクリプトによって制御可能としたものである。

【0026】

より詳細には、各計算機(1)は、図2を参照すると、そのエージェントプラットフォーム(2)が、分散アプリケーションを制御するためのスクリプト言語の入力を受けて前記スクリプト言語を解釈して実行するシェルエージェント(10)と、計算機にローカルなファイルシステムの情報を提供するローカルサービスエージェント(11)と、アプリケーションを直接制御するためのアプリケーションエージェント(12)を持つことのできるエージェントベース(9)と、エージェントを他の計算機上のエージェントベースに移動させるエージェント移動手段(13)と、スレッド生成手段(14)と、エージェントが自計算機又は他の計算機中の他のエージェントと通信を行うリモートコール手段(15)と、アプリケーションエージェントを生成するエージェント生成手段(16)とを備え、入力手段(7)を通して入力されたスクリプト言語を、シェルエージェント(10)が解釈してアプリケーションエージェント(12)を起動し、アプリケーションエージェント(12)が実際のアプリケーション(18)を管理し、シェルエージェント、アプリケーションエージェントは、エージェント移動手段(

1 3) を用いて別の計算機に移動自在とされ、リモートコール手段 (1 5) を用いて別の計算機中のエージェントと通信できる。

【0 0 2 7】

シェルエージェントは、シェルインタプリタ (図 3 の 2 0) と、現在ディレクトリ (図 3 の 2 4) を備えている。

【0 0 2 8】

シェルエージェントは、エージェントのリポジトリパステーブル (図 3 の 3 1) を備え、アプリケーションエージェントを、リポジトリパス上から検索して実行する。

【0 0 2 9】

シェルエージェントのアプリケーションエージェント生成のために、各計算機にローカルな資源へのアクセスを補助するローカルサービスエージェント (図 4 の 1 1) を呼び出す手段 (図 4 の 3 0) を備え、アプリケーションエージェントを生成する時に、シェルエージェントが、直接、エージェント生成手段 (1 6) を呼び出すのではなく、ローカルサービスエージェント (1 1) を介して、アプリケーションエージェントを生成する。

【0 0 3 0】

シェルエージェントは、エージェントの状態を管理する状態テーブル (図 3 の 2 3) を備え、状態テーブルは、エージェントの並列実行カウントと、端末実行フラグと、を備え、前記現在ディレクトリの現在ベースが、リモート計算機である時に、エージェントの状態が並列実行状態でなく、端末実行モードでもないときに、シェルエージェントを、前記リモート計算機に移動させる。

【0 0 3 1】

本発明は、別の実施の形態において、シェルインタプリタが、「parallel」 (パラレル) というキーワードを解釈し、複数のコマンドを並列に実行するために、スレッドの生成、リモート計算機中へのサブシェルエージェント生成を行う。

【0 0 3 2】

より詳細には、シェルインタプリタ (図 3 の 2 0) が、構文解析手段 (図 3 の



21)を備え、スレッド生成手段(図3の14)と、リモートコール手段(図3の13)とを用いて、リモート計算機に、サブシェルエージェント(図18の51)を生成する。

【0033】

入力されたスクリプト言語中に、並列実行を指定する構文(「parallel文」)が存在する場合、前記スクリプト言語を実行するのに必要な分のスレッドを生成して、アプリケーションの並列実行を制御する。

【0034】

エージェントプラットフォームが、生成されたスレッドによるリモート計算機中のアプリケーションエージェントを生成する時に、前記リモート計算機中のエージェントベース中に、サブシェルエージェントを有し、前記サブシェルエージェントが、アプリケーションエージェント生成を代行する。

【0035】

本発明は、さらに別の実施の形態において、起動されるアプリケーションエージェントが、計算機に依存した特定アプリケーションプログラムを実行すしたり、あるいは汎用のアプリケーションエージェントを用いることで、ファイルに依存してアプリケーションを選択的に実行する。

【0036】

より詳細には、アプリケーションエージェント(図24の12)は、特定のアプリケーション(図24の18)に関係付けられるか、あるいは、汎用アプリケーションエージェント(図25の97)として、拡張子アプリケーション対応テーブル(図25の98)を用いて、アプリケーション(図25の18)を決定する。

【0037】

シェルエージェントのシェルインタプリタ(図3の20)は、現在ディレクトリ(図3の24)の情報を参照しながら、スクリプト言語の解釈と実行を行い、必要に応じて、エージェント移動手段(図3の13)によるシェルエージェント自身の移動、リモート計算機との通信を行うためにリモートコール手段(図3の15)、アプリケーションを直接制御するアプリケーションエージェントを生成

するためのエージェント生成手段（図 3 の 16）を呼び出す。

【0038】

シェルインタプリタの構文解析手段（21）は、入力されたスクリプト言語の parallel 文を実行するためにスレッド生成を、スレッド生成手段（14）に依頼し、さらにリモートコール手段（15）を用いて、リモート計算機にサブシェルエージェントを生成し、そのサブシェルエージェントが実際のアプリケーションエージェントをリモート計算機で実行する。

【0039】

アプリケーションエージェントは、ファイルを、特定のアプリケーションを用いて実行するか、あるいは、汎用のアプリケーションエージェントが、計算機ごとに管理された拡張子アプリケーション対応テーブルを用いて、ファイルの名前に依存して、アプリケーションを決定して、実行する。

【0040】

なお、シェルエージェント、ローカルサービス、エージェント、アプリケーションエージェント、エージェント移動手段、スレッド生成手段、リモートコール生成手段、エージェント生成手段、シェルエージェントの構文解析手段、コマンド解析・実行手段等は、計算機上で該当プログラムを実行することによってその機能が実現される。これらのプログラムは、公知の静的又は動的な媒体、例えば記録媒体もしくは通信媒体等から所定のインタフェース手段を介して計算機にロードされ、インストールして実行イメージを実行することで、本発明を実施することができる。

【0041】

【実施例】

上記した本発明の実施の形態についてさらに詳細に説明すべく、本発明の実施例について図面を参照して詳細に説明する。

【0042】

図 1 は、本発明の一実施例のシステム構成を示す図であり、ネットワークスクリプト言語を実行する、ネットワーク接続された複数の計算機よりなる分散処理システムの構成を示す図である。なお、図 1 には、3 台の計算機よりなるシステ

ムが例示されているが、図 1 をはじめ各図は、本発明を説明するためのものであり、本発明を限定するためのものではなく、例えば本発明において、計算機が 3 台の構成に限定されるものでないことは勿論である。

【 0 0 4 3 】

図 1 を参照すると、計算機 1 A、1 B、1 C は、それぞれエージェントプラットフォーム 2 A、2 B、2 C を備え、エージェントプラットフォームは、それぞれエージェントベース 9 A、9 B、9 C を備え、エージェントベース 9 A、9 B、9 C に、それぞれ、エージェント 3 A、3 B、3 C が構築される。

【 0 0 4 4 】

計算機 1 A、1 B、1 C は、ネットワーク 4 で相互に接続されており、ネットワーク 4 を用いて、エージェントプラットフォーム間、あるいは、エージェント間の通信、エージェントの移動等が行われる。

【 0 0 4 5 】

図 2 は、本発明の一実施例における、ネットワークスクリプト言語を実行する計算機の構成の一例を示す図である。

【 0 0 4 6 】

図 2 を参照すると、計算機 1 は、入力 5 を受け、出力 6 を行い、エージェントプラットフォーム 2 と、ローカルファイルシステム 1 7 と、アプリケーション 1 8 と、エージェントリポジトリ 1 9 とを備えている。

【 0 0 4 7 】

エージェントプラットフォーム 2 は、計算機 1 への入力 5 を受ける入力手段 7 と、出力 6 を行う出力手段 8 とを備え、シェルエージェント 1 0 と、ローカルサービスエージェント 1 1 と、アプリケーションエージェント 1 2 等の各種実行中エージェントプログラムを格納する領域として、エージェントベース 9 を備える。

【 0 0 4 8 】

さらに、エージェント移動手段 1 3 と、スレッド生成手段 1 4 と、リモートコール手段 1 5 と、エージェント生成手段 1 6 とを備え、これら各手段はエージェントの実行を補助する機能を具備する。

【0049】

エージェント移動手段13は、エージェントベース9中のエージェントが、他のエージェントプラットフォームに移動する機能を提供する。

【0050】

スレッド生成手段14は、エージェントベース9中のエージェントが、マルチスレッドで動作する時に、新しいスレッドを生成するための機能を提供する。

【0051】

リモートコール手段15は、エージェントベース9中のエージェントが、同じエージェントプラットフォーム中の別のエージェント、あるいは、他のエージェントプラットフォームのエージェントにメソッド呼び出しをする機能を提供する。

【0052】

エージェント生成手段16は、新しいエージェントをエージェントベース9の中に生成し、実行する時に呼び出される手段であり、エージェントリポジトリ19中に、エージェントプログラムを検索し、検索結果に基づいて、エージェントベース9の中に、エージェントを生成する。

【0053】

シェルエージェント10は、ネットワークスクリプト言語を解釈し、実行するエージェントであり、ローカルサービスエージェント11との通信、及び、アプリケーションエージェント12の生成と通信を行う。

【0054】

ローカルサービスエージェント11は、計算機1に固有の情報や機能を提供するエージェントであり、ローカルファイルシステム17を管理してシェルエージェント10に対して情報の提供を行う。

【0055】

アプリケーションエージェント12には2つの互いに異なる型が存在する。

【0056】

このうち第1の型のアプリケーションエージェントは、外部のアプリケーション18をラップする形で存在し、外部のアプリケーション18がエージェントベ

ース 9 内のエージェントと通信や交渉する時の仲介を行う。

【 0 0 5 7 】

第 2 の型のアプリケーションエージェントは、エージェントベース 9 内のエージェントとして、外部のアプリケーションプログラム 1 8 を持たずに、独立に存在して、特定アプリケーションを実行する。

【 0 0 5 8 】

いずれの場合も、アプリケーションエージェント 1 2 は、シェルエージェント 1 0 からの要求で、ある特定のタスクを実行する。

【 0 0 5 9 】

シェルエージェント 1 0、ローカルサービスエージェント 1 1、アプリケーションエージェント 1 2、及びエージェントベース 9、エージェント移動手段 1 3 と、スレッド生成手段 1 4 と、リモートコール手段 1 5 と、エージェント生成手段 1 6 は、計算機 1 上で実行されるプログラムによりその処理・機能が実現される。本発明の一実施例において、これらのプログラムを記録した記録媒体（CD-ROM、FD、DVD、磁気テープ、リムーバブル HDD 等）から、該記録媒体の読み出し装置、読み出し装置のインタフェースを介して、あるいは、サーバ等から有線もしくは無線の通信伝送媒体を介して伝送されたプログラムを通信装置およびそのインタフェースを介して、計算機に読み出し、該プログラムの実行イメージを主記憶にロードし CPU で実行することで、本発明を実施することができる。

【 0 0 6 0 】

図 3 は、本発明の一実施例における、シェルエージェント 1 0 の構成の一例を示す図である。図 3 を参照すると、シェルエージェント 1 0 は、その中心となる構成要素として、シェルインタプリタ 2 0 を備えており、シェルインタプリタ 2 0 は、入力手段 7 からスクリプト言語の入力を受け、入力されたスクリプト言語を構文解析する構文解析手段 2 1 と、コマンドの解析及び実行を行うコマンド解析・実行手段 2 2 と、を備えている。

【 0 0 6 1 】

構文解析手段 2 1 は、スクリプト言語の構文解析にあたり、構文要素のキーワ

ード文字列である「parallel」等のキーワードを管理登録しているキーワードテーブル 2 7 を参照する。

【 0 0 6 2 】

構文解析手段 2 1 は、「parallel」で始まる構文を持つ入力を受理した場合には、スレッド生成手段 1 4 に対して、新しいスレッドの生成を依頼する。

【 0 0 6 3 】

コマンド解析・実行手段 2 2 は、内部コマンドテーブル 2 8 を参照して、入力されたコマンドが、シェルエージェント 1 0 内部で処理可能であるか否かを判定する。

【 0 0 6 4 】

判定の結果、内部コマンドの場合、内部コマンド定義 2 9 からコマンドの定義をロードして実行する。

【 0 0 6 5 】

コマンド解析・実行手段 2 2 は、内部コマンド定義 2 9 からロードされた内部コマンドのうち、ディレクトリ移動のコマンドを実行するにあたり、シェルエージェント 1 0 自身の移動を必要とする場合、エージェント移動手段 1 3 を呼び出す。

【 0 0 6 6 】

また、コマンド解析・実行手段 2 2 は、他のエージェントのメソッドを実行する場合には、リモートコール手段 1 5 を利用する。

【 0 0 6 7 】

コマンド解析・実行手段 2 2 によるコマンド実行の結果は、出力手段 8 に渡されて、計算機 1 から出力される。

【 0 0 6 8 】

一方、外部コマンドの実行は、外部コマンド実行手段 3 0 によって行われる。外部コマンド実行手段 3 0 は、エージェントリポジトリ 1 9 より、外部コマンドを実行するエージェントプログラムを検索する。その際、エージェントリポジトリ 1 9 内のどのパスを検索するか否かを決定するために、リポジトリパステーブル 3 1 を参照し、リポジトリパステーブル 3 1 に格納されたパスについて、順次

エージェントリポジトリ 19 を検索する。

【0069】

外部コマンド実行手段 30 は、エージェントリポジトリ 19 を検索した結果、必要なエージェントが検索された場合、エージェント生成手段 16 に対して、エージェントの生成を依頼し、この依頼を受けたエージェント生成手段 16 が、アプリケーションエージェント 12 を生成する。

【0070】

また、外部コマンド実行手段 30 は、シェルエージェント 10 以外のエージェントにメソッドコールを行う場合、リモートコール手段 15 を呼び出す。

【0071】

シェルエージェント 10 は、シェルエージェント 10 の実行状態を管理する状態テーブル 23 と、シェルエージェント 10 の現在ディレクトリ 24 と、シェル変数を管理するシェル変数管理テーブル 25 と、外部のエージェントを指し示す変数を管理するエージェント参照テーブル 26 と、を備え、これらのテーブルはシェルインタプリタ 20 の実行を補助する。

【0072】

シェル変数管理テーブル 25 は、シェル変数によってスクリプト言語を記述し易くするための管理テーブルである。なお、シェル変数管理テーブル 25 を具備しない構成で、図 3 のシェルエージェント 10 を構成するようにしてもよい。

【0073】

図 4 は、本発明の一実施例の変形として、シェルエージェント 10 を構成する要素について、図 3 に示したものと別の構成を示す図である。図 3 に示した構成では、外部コマンド実行手段 30 は、直接エージェント生成手段 16 を呼び出しているが、この機能を、ローカルサービスエージェント 11 に依頼するようにしてもよい。

【0074】

この場合、このローカルサービスエージェント 11 が、外部コマンド実行手段 30 から、外部コマンド名と、リポジトリパステーブル 31 に格納されたパス情報とを受け取り、エージェントリポジトリ 19 の検索と、実際のエージェント生

成手段 16 の呼び出しを行うことで、アプリケーションエージェント 12 を生成する。

【0075】

シェルエージェント 0 のシェルインタプリタ 21 の構文解析手段 21、コマンド解析・実行手段 22、外部コマンド実行手段 30 は、計算機 1 上で実行されるプログラムによりその処理・機能が実現される。

【0076】

図 5 は、本発明の一実施例において、現在ディレクトリ 24 を含めたディレクトリ管理を行う情報の格納テーブルの構成の一例を示す図である。

【0077】

図 5 を参照すると、ディレクトリは、属性名 32 と値 33 との対で管理されており、属性として、シェルエージェント 10 が起動されたエージェントベース 9 を表すルートベース 34 と、このディレクトリが存在する現在ベース 35 と、ベース内のローカルなディレクトリを示すベース内ディレクトリ 36 と、からなる。

【0078】

例えば、ベース A で立ち上がったシェルが、ベース B の「/home」というディレクトリを指し示す場合には、ルートベースは、ベース A のベース、現在ベースは、ベース B のベース、ベース内ディレクトリには、「/home」が格納される。

【0079】

利用者によるディレクトリの指定方法は、絶対パスの指定の仕方として、

- (1) 「//ベース名A/ベース内ディレクトリA」、あるいは、
- (2) 「///ベース内ディレクトリB」

となる。

【0080】

(1) の場合、現在ベースは「ベース名A」、ベース内ディレクトリは、「ベース内ディレクトリA」である。(2) の場合、現在ベースは「ルートベース」、ベース内ディレクトリは、「ベース内ディレクトリB」となる。

【0081】



この他に、相対的なディレクトリを表現するものとして、現在ディレクトリから相対的に一つ上のディレクトリを指し示す「../」、現在ディレクトリの下「dirA」というディレクトリの下「dirB」というディレクトリを示す「dirA/dirB」などの表現方法を用いてもよい。

## 【 0 0 8 2 】

図 6 は、本発明の一実施例における状態テーブル 2 3 の構成の一例を示す図である。図 6 を参照すると、状態は、属性名 3 7 と値 3 8 との対で管理されており、属性として、シェルエージェント 1 0 が並列実行中であるかどうかを示す並列実行カウンタ 3 9 と、シェルエージェント 1 0 が端末入力モードで実行されているのか、バッチモードで実行されているかを示す端末モードフラグ 4 0 と、からなる。

## 【 0 0 8 3 】

図 7 は、本発明の一実施例におけるシェル変数管理テーブル 2 5 の構成の一例を示す図である。図 7 を参照すると、シェル変数管理テーブル 2 5 は、シェル変数名 4 1 と値 4 2 の対でシェル変数を管理しており、シェル変数の数だけ変数管理単位 4 3 を持つ。

## 【 0 0 8 4 】

図 8 は、本発明の一実施例において、シェルエージェント 1 0 の中で、エージェントを指し示す変数であるエージェント変数の管理を行うエージェント参照テーブル 2 6 の構成の一例を示す図である。図 8 を参照すると、エージェント参照テーブル 2 6 は、エージェント変数名 4 4 と外部エージェント 4 5 の対でエージェント変数を管理しており、エージェント変数の数だけ変数管理単位 4 6 を持つ。

## 【 0 0 8 5 】

図 9 は、本発明の一実施例において、シェルエージェント 1 0 のシェルインタプリタ 2 0 の動作を説明するための流れ図である。

## 【 0 0 8 6 】

まず、シェルインタプリタ 2 0 は、入力手段 7 からコマンド入力を受け取る（ステップ 5 2）。

【 0 0 8 7 】

次に、このコマンドが終了コマンドかどうかを判断し（ステップ 5 3）、終了コマンドであれば、終了処理を実行し（ステップ 5 7）、シェルエージェント 1 0 を終了する。

【 0 0 8 8 】

終了コマンドでなければ、キーワードテーブル 2 7 を用いて構文解析手段 2 1 で構文解析し、parallel文であるか否かを判断する（ステップ 5 4）。

【 0 0 8 9 】

parallel文である場合には、シェルエージェント 1 0 は並列でコマンド実行を行うことになるので、ステップ 5 8 の並列実行処理へ進む。

【 0 0 9 0 】

parallel文でない場合には、コマンド解析・実行手段 2 2 により、内部コマンドテーブル 2 8 を用いて内部コマンドとして処理できるか否かを判断する（ステップ 5 5）。

【 0 0 9 1 】

内部コマンドの場合には、ステップ 5 9 の内部コマンド処理に進み、内部コマンド定義 2 9 を用いて、内部コマンドの実行を行う。

【 0 0 9 2 】

内部コマンドでない場合には、外部コマンド処理を行う（ステップ 5 6）。

【 0 0 9 3 】

並列実行処理（ステップ 5 8）、内部コマンド処理（ステップ 5 9）、外部コマンド処理（ステップ 5 6）をそれぞれ終了した場合、ステップ 5 2 のコマンド入力処理に戻り、処理が継続される。

【 0 0 9 4 】

図 9 の終了コマンドであるか否かの判断（ステップ 5 3）と、構文解析してparallel文かどうかを判断する処理（ステップ 5 4）の順序は、逆であっても良い。

【 0 0 9 5 】

図 1 0 は、本発明の一実施例において、シェルエージェント 1 0 のシェルイン

タブリタ 20 のコマンド解析・実行手段 22 における内部コマンド処理（図 9 のステップ 59）の処理手順を示す流れ図である。

【0096】

まず、内部コマンドがディレクトリ移動を表すコマンド「cd」（cd コマンド）であるかどうかを判断する（ステップ 67）。

【0097】

「cd」でないコマンドは、内部コマンド定義 29 に格納された内部コマンド定義に従って実行する（ステップ 73）。

【0098】

内部コマンドが「cd」である場合、図 11 に示すように、まず、移動先ディレクトリ 47 を、cd コマンドから抽出する。cd コマンドは、移動先ディレクトリが指定される。

【0099】

この実施例では、利用者は、ディレクトリを「//ベース名/ベース内のディレクトリ」の形で指定するため、これを、図 5 に示したディレクトリ構成要素に展開し、ベース名を現在ベース 35 に、ベース内のディレクトリをベース内ディレクトリ 36 にそれぞれ格納する。

【0100】

次に、状態テーブル 23 に設けられた並列実行カウンタ 39 の値が「0」であるか否かを検査する（ステップ 68）。

【0101】

この並列実行カウンタには、現在シェルエージェントがマルチスレッドで動作しているかどうかを表す情報として、並列実行の深さ情報が格納されている。

【0102】

並列実行されていなければ、並列実行カウンタの値は、「0」であり、一重の parallel 構文の中であれば「1」であり、parallel 構文の中の parallel 構文の中であれば「2」となる。

【0103】

並列実行カウンタの値が「0」以外の時は、シェルエージェントが並列実行し

ていることを表しており、シェルエージェント全体が別の計算機に移動する価値が低いので、現在ディレクトリ 24 の値を、cd コマンドで示された移動先ディレクトリの値に更新する処理（ステップ 72）だけを行い、シェルエージェント自身を計算機間で移動させない。

**【0104】**

並列実行カウンタの値が「0」である場合には、シェルエージェントは単スレッドで実行しているので、次の端末モードの検査を行う（ステップ 69）。

**【0105】**

状態テーブル 23 に設けられた端末モードフラグ 40 は、シェルエージェントが端末から入力待つシェルとして実行されている時は「1」、バッチコマンドとして特定の端末に接続されないで実行されている時は「0」に、その値が設定される。

**【0106】**

端末モードフラグ 40 の値が「1」であるということは、端末入力を待っているシェルエージェントであることから、シェルエージェントが別の計算機に移動してしまうと入力応答速度が遅くなるため、cd コマンドによるシェルエージェントの計算機間移動は行わず、ステップ 72 に分岐して、現在ディレクトリの値の更新だけを行う。

**【0107】**

端末モードフラグ 40 の値が「0」の時は、バッチモードの実行のため、シェルエージェントは、作業ディレクトリのある計算機に移動して実行する方が有利である。そこで、現在ディレクトリ 24 と、移動先ディレクトリ 47 の値の比較を行う（ステップ 70）。

**【0108】**

現在ディレクトリ 24 と移動先ディレクトリ 47 が同じ計算機上にある場合（現在ベース 35 の値が同じ場合）には、シェルエージェントの計算機間移動は不要のため、ステップ 72 へ分岐して、そのまま現在ディレクトリの値の更新だけを行う。

**【0109】**

現在ディレクトリ 2 4 と移動先ディレクトリ 4 7 が同じ計算機にない場合（現在ベース 3 5 の値が異なる場合）には、シェルエージェントを移動先ディレクトリの現在ベースが存在する計算機のエージェントベースに移動する（ステップ 7 1）。

【 0 1 1 0 】

このように、利用者の発行する「cd」という内部コマンドは、計算機内のディレクトリ移動についての処理を行うと同時に、シェルエージェント内部では、シェルエージェントを実行する計算機を選択にも利用される。

【 0 1 1 1 】

かかる構成によって、リモート計算機上のアプリケーションの実行であっても、利用者には、ディレクトリの移動とエージェントの実行として見せることができることで、ネットワークに分散したサービスを実行するために複数のリモートアプリケーションを連携させる場合に、簡単なスクリプト言語で、記述することができる。

【 0 1 1 2 】

図 1 2 及び図 1 3 は、本発明の一実施例において、シェルエージェントを移動する時の構成の変化を説明するための図である。

【 0 1 1 3 】

まず、図 1 2 を参照すると、エージェントプラットフォーム A 2 A 上にいるシェルエージェント 1 0 は、エージェント移動手段 1 3 A に対して、自身の移動を依頼する。

【 0 1 1 4 】

エージェント移動手段 1 3 A は、エージェントプラットフォーム B 2 B のエージェント移動手段 1 3 B に、シェルエージェント 1 0 の状態を転送する。

【 0 1 1 5 】

そして、図 1 3 を参照すると、エージェント移動手段 1 3 B は、エージェントプラットフォーム B 2 B 上にシェルエージェント 1 0 を再構築して実行を再開する。

【 0 1 1 6 】

図 14 は、本発明の一実施例において、コマンド解析・実行手段 22 において内部コマンドを実行するにあたり、内部コマンドが、外部のアプリケーションエージェント 12 へのメソッド実行や、移動要求を出す場合を説明するための図である。

## 【0117】

まず、コマンド解析・実行手段 22 は、エージェント変数の指し示す外部のアプリケーションエージェントを同定するために、エージェント参照テーブル 26 を用いて、エージェント変数名から外部のアプリケーションエージェントへのポインタを獲得する。

## 【0118】

そして、コマンド解析・実行手段 22 は、リモートコール手段 15 に対して、外部アプリケーションエージェント 12 へのメソッド実行や移動を要求する。

## 【0119】

アプリケーションエージェント 12 は、リモートコールの受け入れ手段として、移動受入手段 48 と命令受入手段 49 とを有する。

## 【0120】

移動受入手段 48 は、移動要求の権限について確認をした後に、エージェント移動手段 13 に対して、アプリケーションエージェント 12 自身の移動を要求する。

## 【0121】

命令受入手段 49 は、命令実行の権限について確認した後に、命令実行手段 50 を実行する。

## 【0122】

図 15 及び図 16 は、本発明の一実施例において、シェルエージェント 10 からの同じ計算機中のアプリケーションエージェント 12 を、別の計算機へ移動する時の構成の変化を説明するための図である。

## 【0123】

まず、図 15 を参照すると、シェルエージェント 10 は、リモートコール手段 15 を用いて、アプリケーションエージェント 12 に移動の命令を伝える。

## 【0124】

アプリケーションエージェント 12 は、エージェント移動手段 13 A に対して自身の移動を依頼し、このエージェント移動手段 13 A は、エージェントプラットフォーム B 2 B のエージェント移動手段 13 B に、アプリケーションエージェントの状態を転送する。

## 【0125】

そして、図 16 を参照すると、エージェント移動手段 13 B は、エージェントプラットフォーム B 2 B 上にアプリケーションエージェント 12 を再構成し、実行を再開する。

## 【0126】

図 15 及び図 16 では、シェルエージェント 10 と同じ第 1 のエージェントプラットフォームのアプリケーションエージェント 12 を移動したが、アプリケーションエージェントは、シェルエージェントとは異なる第 2 のエージェントプラットフォーム中に存在して、これを第 3 のエージェントプラットフォーム、あるいはシェルエージェントの存在する第 1 のエージェントプラットフォームに移動することもできる。

## 【0127】

図 17 は、本発明の一実施例において、構文解析手段 21 において、parallel 文が検出され、図 9 の並列実行処理（ステップ 58）を実行する処理手順を示す流れ図である。

## 【0128】

まず、状態テーブル 23 の並列実行カウンタ 39 に 1 を加える（ステップ 60）。これにより、シェルエージェントが現在並列実行中であることが記録される。

## 【0129】

次に、parallel 文中のコマンド数分だけのスレッドを生成する（ステップ 66）。スレッドの生成は、スレッド生成手段 14 によって行われる。

## 【0130】

続いて、変数「threads」（スレッド）に、生成したスレッド数を設定する（

ステップ 6 2)。

【0 1 3 1】

そして、実際に、parallel文の中のコマンドを各スレッドで並列に実行する（ステップ 6 3）。

【0 1 3 2】

各スレッドは、各自の実行が終了すると、threads変数の値を「1」だけ減少させる（ステップ 6 4）。この処理は、スレッド間で排他制御されていなければならない。

【0 1 3 3】

そして、threads変数が「0」になったか否かを判断し（ステップ 6 5）、0でない場合には、次のスレッドが終了するのを待つ。

【0 1 3 4】

全てのスレッドが終了すると、threads変数の値は「0」になり、並列実行は終了する。

【0 1 3 5】

そして、並列実行カウンタ 3 9 の値を 1 デクリメント（減算）する処理を行い（ステップ 6 6）、並列実行処理全体の実行を終了する。

【0 1 3 6】

図 1 8 は、本発明の一実施例において、parallel文の中で、特に、生成されたスレッドが別のエージェントプラットフォームに対してコマンドを実行する時の構成を示す図である。

【0 1 3 7】

図 1 8 を参照すると、シェルエージェント 1 0 は、スレッド生成手段 1 4 に、新しいスレッド生成を依頼した後、parallel文の内部コマンド実行のために、リモートコール手段 1 5 A、1 5 B を用いて、サブシェルエージェント 5 1 を、エージェントプラットフォーム 2 B のエージェントベース 9 B 上に作成する。

【0 1 3 8】

以後、このサブシェルエージェント 5 1 が、アプリケーションエージェント 1 2 や、ローカルサービスエージェント B 1 1 B に対して、メソッド実行や移動



命令を実行することができる。

【 0 1 3 9 】

図 1 9 は、本発明の一実施例において、parallel文の中で複数のエージェントプラットフォームに対して、並列にコマンド実行を行う場合を説明するための図である。

【 0 1 4 0 】

図 1 9 を参照すると、シェルエージェント 1 0 からのリモートコールによって、サブシェルエージェント 5 1 B、サブシェルエージェント 5 1 C が生成され、2つのエージェントプラットフォーム 2 B、2 C の上で並列にサブシェルエージェントが実行を行うことができる。

【 0 1 4 1 】

図 2 0 は、本発明の一実施例において、図 9 の外部コマンド処理（ステップ 5 6）を実行するための処理手順を示す流れ図である。

【 0 1 4 2 】

図 2 0 を参照すると、外部のアプリケーションエージェントを起動する方法として、図 3 に示したように、直接、外部コマンド実行手段 3 0 がエージェント生成手段 1 6 を呼び出す方法を用いる。

【 0 1 4 3 】

まず、現在ディレクトリ 2 4 に格納されている現在ベースが、実際に、シェルエージェントを実行している実行中ベースと一致しているかを判断する（ステップ 7 4）。

【 0 1 4 4 】

一致している場合には、実行中ベースに外部のアプリケーションエージェントを生成するため、まず、実行中ベースのエージェントリポジトリ 1 9 のエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断する（ステップ 7 5）。

【 0 1 4 5 】

アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリ 1 9 からアプリケーションエージェントのプログラムをロードしてアプリ

ケーションエージェント 12 を実行する（ステップ 76）ように、エージェント生成手段 16 に対して依頼する。

【0146】

エージェントリポジトリ 19 に見つからない場合には、実行に失敗する（ステップ 80）。

【0147】

現在ディレクトリ 24 に格納されている現在ベースと、実際にシェルエージェントを実行している実行中ベースと一致しているかを判定するステップ 74 において、一致していない場合には、リモートコール手段 15 を用いて、現在ベース中にサブシェルエージェント 51 を作成する（ステップ 77）。

【0148】

図 21 は、この場合の構成要素の関係を示す図である。

【0149】

サブシェルエージェント 51 中で現在ベースのエージェントリポジトリ 19 のエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するかを判断する（ステップ 87）。

【0150】

アプリケーションエージェントのプログラムがあれば、エージェントリポジトリ 19 からアプリケーションエージェントのプログラムをロードしてアプリケーションエージェント 12 を実行する（ステップ 79）ようにエージェント生成手段 16 に依頼する。

【0151】

エージェントリポジトリに見つからない場合は、実行に失敗する（ステップ 80）。

【0152】

図 22 は、本発明の一実施例において、図 9 の外部コマンド処理（ステップ 56）を実行するための処理手順を示す流れ図である。

【0153】

図 22 を参照すると、外部のアプリケーションエージェントを起動する方法と

して、図4に示したようにローカルサービスエージェント11にエージェント生成に関するタスクを委任してこのローカルサービスエージェント11がエージェント生成手段16を呼び出す方法を用いる。

【0154】

まず、現在ディレクトリ24に格納されている現在ベースが、実際にシェルエージェントを実行している実行中ベースと一致しているか否かを判断する（ステップ81）。

【0155】

一致している場合には、実行中ベースに外部のアプリケーションエージェントを生成するため、まず、実行中ベースのローカルサービスエージェント11に外部エージェントの実行依頼を行う（ステップ82）。

【0156】

次に、このローカルサービスエージェント11は、実行中ベースのエージェントリポジトリ19のエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断する（ステップ83）。

【0157】

アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリ19からアプリケーションエージェントのプログラムをロードしてアプリケーションエージェント12を実行するように（ステップ84）、エージェント生成手段16に対して依頼する。

【0158】

エージェントリポジトリに見つからない場合は、実行に失敗する（ステップ89）。

【0159】

現在ディレクトリ24に格納されている現在ベースと、実際にシェルエージェントを実行している実行中ベースと一致しているか否かを判断するステップ81で、一致していない場合には、リモートコール手段15を用いて、現在ベース中のローカルサービスエージェント11に外部エージェントの実行依頼を行う（ステップ86）。

【0160】

図23は、この場合の各構成要素の関係を示す図である。

【0161】

図23を参照すると、このローカルサービスエージェント11Bは、現在ベースのエージェントリポジトリ19のエージェントリポジトリパス中に対象のアプリケーションエージェントのプログラムが存在するか否かを判断する（図2のステップ87）。

【0162】

アプリケーションエージェントのプログラムが存在すれば、エージェントリポジトリ19からアプリケーションエージェントのプログラムをロードしてアプリケーションエージェント12を実行するように（図22のステップ88）、エージェント生成手段16に依頼する。

【0163】

エージェントリポジトリに見つからない場合は、実行に失敗する（図22のステップ89）。

【0164】

図24は、本発明の一実施例において、アプリケーションエージェント12と指定されたファイル96、アプリケーションエージェントが起動する外部アプリケーション18の関係を示す図である。

【0165】

アプリケーションエージェント12は、特定の外部アプリケーション18と密接に関連づけられてファイル96を解釈実行することができる。

【0166】

図25は、本発明の一実施例において、汎用アプリケーションエージェント97を利用して、計算機に依存したアプリケーション12を起動するための構成を示す図である。

【0167】

図25を参照すると、汎用アプリケーションエージェント97は、指定されたファイル96の拡張子を見て、拡張子アプリケーション対応テーブル98を参照

することで、実行すべきアプリケーションを決定する。この拡張子アプリケーション対応テーブル 98 は、拡張子 99 とアプリケーション 100 とを対として格納したテーブルであり、拡張子単位の情報 101 が格納されている。

【0168】

汎用アプリケーションエージェント 97 は、ファイル 96 の拡張子と一致した拡張子 99 を持つ項目を、拡張子アプリケーション対応テーブル 98 から検索し、該項目のアプリケーション欄 100 のアプリケーションを起動する。

【0169】

拡張子アプリケーション対応テーブル 98 は、各計算機ごとに具備することができる。このため、例えば、同じ拡張子を持つ画像ファイルであっても、別の計算機では別のアプリケーションを用いて表示させることができる。

【0170】

次に、上記した実施例をさらに具体的に説明するため、本発明を適用した具体例についていくつか説明する。

【0171】

図 26 は、本発明を適用したシステム構成の一例を示す図であり、2つの計算機上で、2つのプレゼンテーションツールを順次制御するシステムの構成を示す図である。

【0172】

まず、計算機 A 1A のエージェントプラットフォーム 2A のエージェントベース 9A にシェルエージェント 10A があり、そこに実行スクリプトファイル 92 を入力する。

【0173】

この場合、実行は、バッチモードで行われるため、状態テーブル 23 の端末モードフラグ 40 は 0 になる。並列実行を示す parallel 文も用いられていないので、スクリプト中の cd コマンドによって、シェルエージェントは、計算機中を移動することになる。

【0174】

スクリプトファイル 92 の 1 行目の「cd」によって、シェルエージェント 10

Aは、計算機B 1 Bのエージェントプラットフォーム 2 Bのエージェントベース 9 Bのローカルディレクトリ「/home」に移動する。

【0 1 7 5】

そこで、スクリプトファイル 9 2の2行目の「ppt」コマンドが実行されるが、ここで、pptコマンドは、外部のプレゼンテーションツール 9 0を起動するアプリケーションエージェント B 1 2 Bの名前であるとする。

【0 1 7 6】

従って、シェルエージェント 1 0 Bは、アプリケーションエージェント B 1 2 Bを起動し、これがプレゼンテーションツール B 9 0を起動してプレゼンテーションを開始する。

【0 1 7 7】

続いて、スクリプトファイル 3 行目のcdコマンドで、シェルエージェント 1 0 Bは、計算機C 1 Cのエージェントプラットフォーム 2 Cのエージェントベース 9 Cのローカルディレクトリ「/home」に移動する。

【0 1 7 8】

そして、4 行目で、アプリケーションエージェント C 1 2 Cを起動し、プレゼンテーションツール C 9 1の実行を開始する。

【0 1 7 9】

再びスクリプトファイル 9 2の5行目の「cd」で、シェルエージェントは、計算機Bに移動し、プレゼンテーションツールBに、「次頁」を表示するためのコマンド「next」を送信する。

【0 1 8 0】

7 行目と 8 行目では、さらに、計算機Cに移動して、プレゼンテーションツールCも次頁を表示する。

【0 1 8 1】

このように、複数計算機のプレゼンテーションツールを、順次ページめくりする動作をひとつのシェルスクリプト中に記述できる。

【0 1 8 2】

図 2 7は、本発明を適用したシステムの別の例を示す図であり、2つの計算機

上で、2つのプレゼンテーションツールを並列制御するシステムの構成を示す図である。

【0 1 8 3】

図 2 7 に示したシステムが、図 2 6 に示したシステムと相違する点は、スクリプトファイル 9 3 中に、parallel の構文が利用されていることである。

【0 1 8 4】

parallel 文の中では、サブシェルエージェントが計算機 B、計算機 C に作成され、それぞれで並列にアプリケーションエージェント B 1 2 B とプレゼンテーションツール B 9 0、あるいはアプリケーションエージェント C 1 2 C とプレゼンテーションツール 9 1 を制御する。

【0 1 8 5】

図 2 6 に示したシステムとは相違して、アプリケーションの操作が並列に行われ、また parallel 文の文末の「end」の行では、「parallel」とされたサブエージェントの同期が行われる。

【0 1 8 6】

図 2 8 は、本発明を適用したシステムのさらに別の例を示す図であり、2つの計算機上で、プレゼンテーションツール 9 0 と、WWW (World Wide Web) ブラウザ 9 5 を並列制御するシステムの構成を示す図である。図 2 7 との違いは、制御対象が、異なるアプリケーションとされていることである。

【0 1 8 7】

この例を用いれば、プレゼンテーションツールと、関連した WWW ページの情報を並列に、同期機能を持って複数の計算機に表示するシステムを容易に構成することができる。

【0 1 8 8】

【発明の効果】

以上説明したように、本発明によれば下記記載の効果を奏する。

【0 1 8 9】

本発明の第 1 の効果は、一つのスクリプトプログラムを用いて、複数の計算機に分散されたアプリケーションの実行を制御することができる、ということであ

る。この結果、複数のアプリケーションを連携させることによる上位の分散アプリケーションを容易に構築できる。

【0190】

その理由は、本発明においては、スクリプト言語の解釈を、移動可能で、かつリモート計算機と通信を行えるシェルエージェントで構築し、アプリケーションの実行を制御するアプリケーションエージェントを、どの計算機上にも自在に生成して実行可能としたためである。

【0191】

本発明の第2の効果は、逐次的なスクリプトの実行だけでなく、並列的な実行が可能であり、さらに、複数の計算機に分散したアプリケーションの並列実行や、同期的な実行を、スクリプト言語を用いて記述することを容易化している、ということである。

【0192】

その理由は、本発明においては、スクリプト言語の構文要素にparallelというブロックを設け、このブロックの中のコマンドを並列に実行するために、複数のスレッド生成と、生成されたスレッドをリモート計算機中のサブシェルエージェントの中で実行させる機能を具備している、ためである。

【0193】

本発明の第3の効果は、計算機間の移動も、計算機内のディレクトリ移動も一つのディレクトリ移動コマンドで行うことができ、特別なエージェント移動というコマンドを用いることなしに、シェルエージェント移動を可能としている、ということである。

【0194】

その理由は、本発明においては、シェルエージェントの中で、現在のディレクトリや、実際にいる計算機、シェルエージェントの状態などを管理し、必要な場合を判断して、シェルエージェントの計算機間の移動を行う機能を装備したためである。

【0195】

本発明の第4の効果は、計算機ごとに用意されたアプリケーションが異なる場



合であっても、ファイルの内容に応じて、アプリケーションを実行することができる、ということである。

【0 1 9 6】

その理由は、本発明においては、アプリケーションエージェントが計算機に応じたアプリケーションを特定して実行できる機能を具備し、汎用のアプリケーションエージェントを作成して、実行するファイルの拡張子に合わせて計算機に依存したアプリケーションを選択的に実行可能としたためである。

【図面の簡単な説明】

【図 1】

本発明の一実施例のシステム構成を示す図である。

【図 2】

本発明の一実施例においてエージェントプラットフォームの構成を示す図である。

【図 3】

本発明の一実施例におけるシェルエージェントの構成の一例を示す図である。

【図 4】

本発明の一実施例におけるシェルエージェントの構成の別の例を示す図である。

【図 5】

本発明の一実施例におけるディレクトリ管理テーブルの構成の一例を示す図である。

【図 6】

本発明の一実施例における状態テーブルの構成の一例を示す図である。

【図 7】

本発明の一実施例におけるシェル変数管理テーブルの構成の一例を示す図である。

【図 8】

本発明の一実施例におけるエージェント参照テーブルの構成の一例を示す図である。

【図 9】

本発明の一実施例においてスクリプト言語を解釈する処理手順を示す流れ図である。

【図 1 0】

本発明の一実施例において内部コマンドを解釈する処理手順を示す流れ図である。

【図 1 1】

本発明の一実施例におけるディレクトリ移動を説明するための図である。

【図 1 2】

本発明の一実施例におけるディレクトリ移動の一例を説明するための図（その 1）である。

【図 1 3】

本発明の一実施例におけるディレクトリ移動の一例を説明するための図（その 2）である。

【図 1 4】

本発明の一実施例におけるリモート命令実行を説明するための図である。

【図 1 5】

本発明の一実施例におけるリモート命令実行の一例を説明するための図（その 1）である。

【図 1 6】

本発明の一実施例におけるリモート命令実行の一例を説明するための図（その 2）である。

【図 1 7】

本発明の一実施例における並列実行を解釈する処理手順を示す流れ図である。

【図 1 8】

本発明の一実施例における並列実行の一例を説明するための図（その 1）である。

【図 1 9】

本発明の一実施例における並列実行の一例を説明するための図（その 2）であ

る。

【図 20】

本発明の一実施例においてアプリケーションエージェントの実行を解釈する処理手順を示す流れ図である。

【図 21】

本発明の一実施例においてアプリケーションエージェントの実行の一例を説明するための図（その 1）である。

【図 22】

本発明の一実施例においてアプリケーションエージェントの実行を解釈する処理手順を示す流れ図である。

【図 23】

本発明の一実施例においてアプリケーションエージェントの実行の一例を説明するための図（その 2）である。

【図 24】

本発明の一実施例におけるアプリケーションエージェントとアプリケーションの関係の一例を示す図である。

【図 25】

本発明の一実施例における汎用アプリケーションエージェントとアプリケーションの関係の一例を示す図である。

【図 26】

本発明を適用したシステムの第 1 の実施例を説明する図である。

【図 27】

本発明を適用したシステムの第 2 の実施例を説明する図である。

【図 28】

本発明を適用したシステムの第 3 の実施例を説明する図である。

【符号の説明】

- 1 計算機
- 2 エージェントプラットフォーム
- 3 エージェント

- 4 ネットワーク
- 5 入力
- 6 出力
- 7 入力手段
- 8 出力手段
- 9 エージェントベース
- 1 0 シェルエージェント
- 1 1 ローカルサービスエージェント
- 1 2 アプリケーションエージェント
- 1 3 エージェント移動手段
- 1 4 スレッド生成手段
- 1 5 リモートコール手段
- 1 6 エージェント生成手段
- 1 7 ローカルファイルシステム
- 1 8 アプリケーション
- 1 9 エージェントリポジトリ
- 2 0 シェルインタプリタ
- 2 1 構文解析手段
- 2 2 コマンド解析・実行手段
- 2 3 状態テーブル
- 2 4 現在ディレクトリ
- 2 5 シェル変数管理テーブル
- 2 6 エージェント参照テーブル
- 2 7 キーワードテーブル
- 2 8 内部コマンドテーブル
- 2 9 内部コマンド定義
- 3 0 外部コマンド実行手段
- 3 1 リポジトリバステーブル
- 3 2 属性名欄

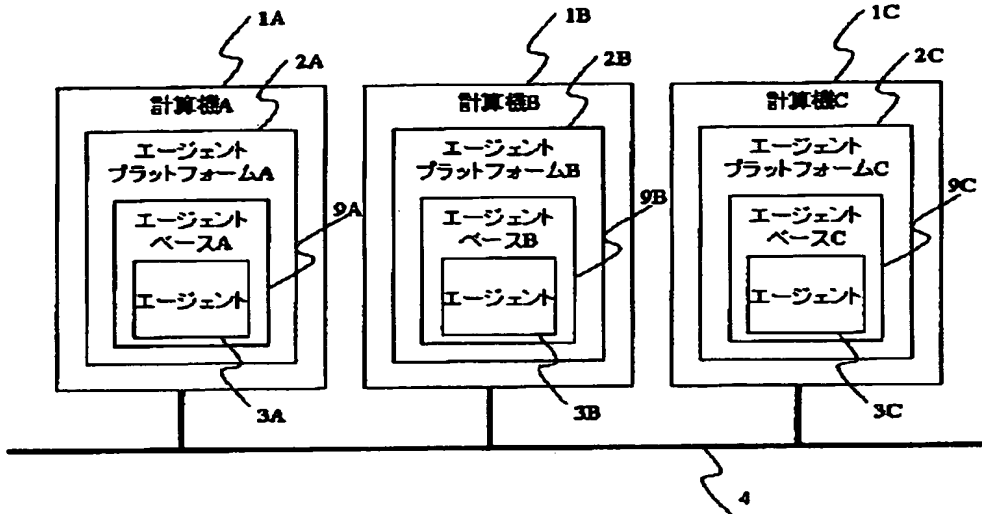
- 3 3 値欄
- 3 4 ルートベース
- 3 5 現在ベース
- 3 6 ベース内ディレクトリ
- 3 7 属性名欄
- 3 8 値欄
- 3 9 並列実行カウンタ
- 4 0 端末モードフラグ
- 4 1 シェル変数名欄
- 4 2 値欄
- 4 3 シェル変数管理単位
- 4 4 エージェント変数名
- 4 5 外部エージェント
- 4 6 エージェント変数管理単位
- 4 7 移動先ディレクトリ
- 4 8 移動受入手段
- 4 9 命令受入手段
- 5 0 命令実行手段
- 5 1 サブシェルエージェント
- 5 2 コマンド入力
- 5 3 終了判断
- 5 4 p a r a l l e l 文判断
- 5 5 内部コマンド判断
- 5 6 外部コマンド処理
- 5 7 終了処理
- 5 8 並列実行処理
- 5 9 内部コマンド処理
- 6 0 並列カウンタのカウントアップ
- 6 1 スレッド生成

- 6 2 スレッド数の設定
- 6 3 並列実行
- 6 4 スレッド終了処理
- 6 5 全スレッドの終了判断
- 6 6 並列カウンタのカウントダウン
- 6 7 c d であるか判断
- 6 8 並列実行中かを判断
- 6 9 端末モードか判断
- 7 0 現在ディレクトリと移動先ディレクトリの比較
- 7 1 シェルエージェント移動
- 7 2 現在ディレクトリの更新
- 7 3 c d 以外の内部コマンドの実行
- 7 4 ベースの一致判断
- 7 5 エージェントプログラムの検索
- 7 6 エージェント実行
- 7 7 サブシェルエージェント作成
- 7 8 エージェントプログラムの検索
- 7 9 エージェント実行
- 8 0 サブシェルエージェント作成
- 8 1 ベースの一致判断
- 8 2 ローカルエージェントへのコマンド依頼
- 8 3 エージェントプログラムの検索
- 8 4 エージェント実行
- 8 6 ローカルエージェントへのコマンド依頼
- 8 7 エージェントプログラムの検索
- 8 8 エージェント実行
- 8 9 サブシェルエージェント作成
- 9 0 プレゼンテーションツール
- 9 1 プレゼンテーションツール

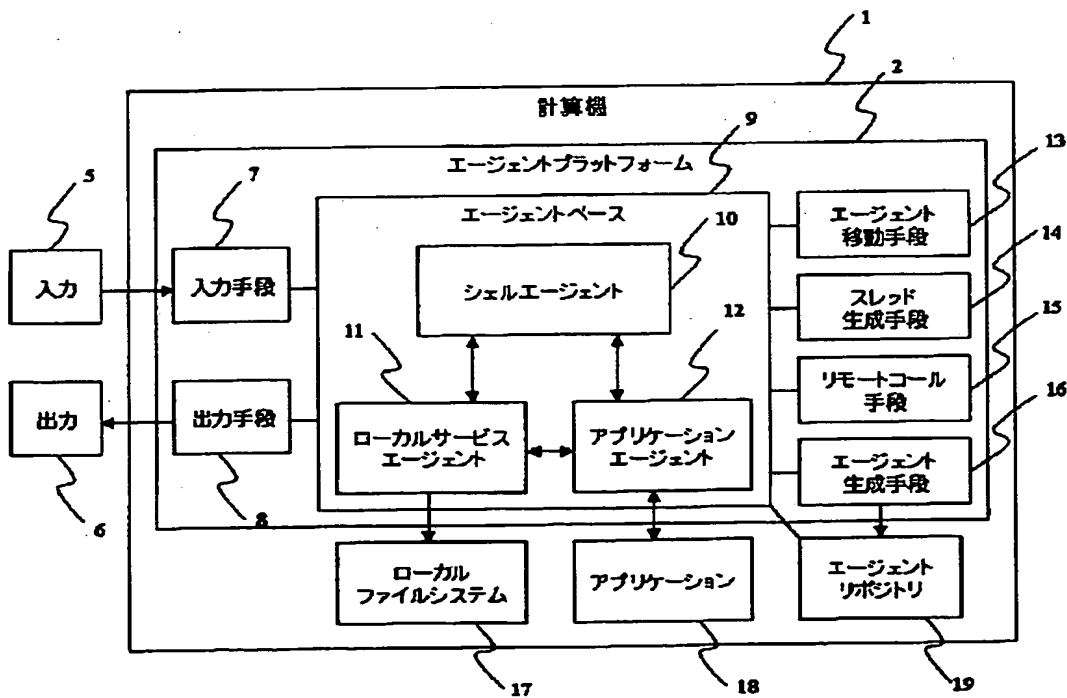
- 9 2 スクリプトプログラム
- 9 3 スクリプトプログラム
- 9 4 スクリプトプログラム
- 9 5 WWWブラウザ
- 9 6 ファイル
- 9 7 汎用アプリケーションエージェント
- 9 8 拡張子アプリケーション対応テーブル
- 9 9 拡張子欄
- 1 0 0 アプリケーション欄
- 1 0 1 対応テーブルの要素

【書類名】 図面

【図 1】

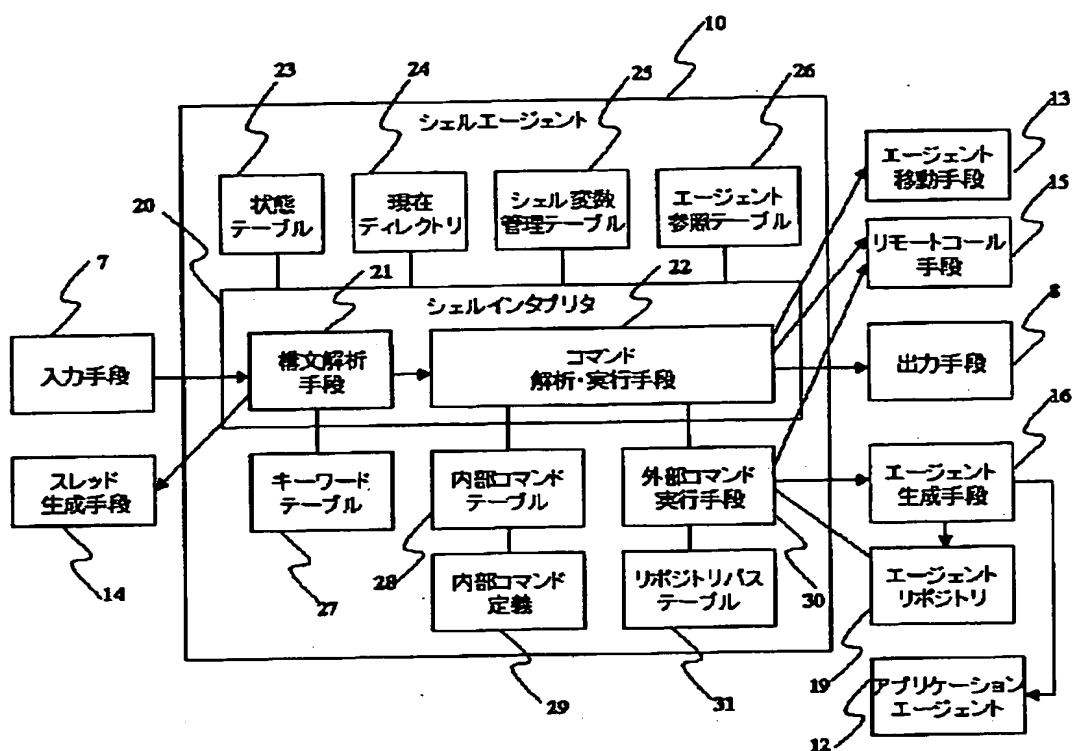


【図 2】

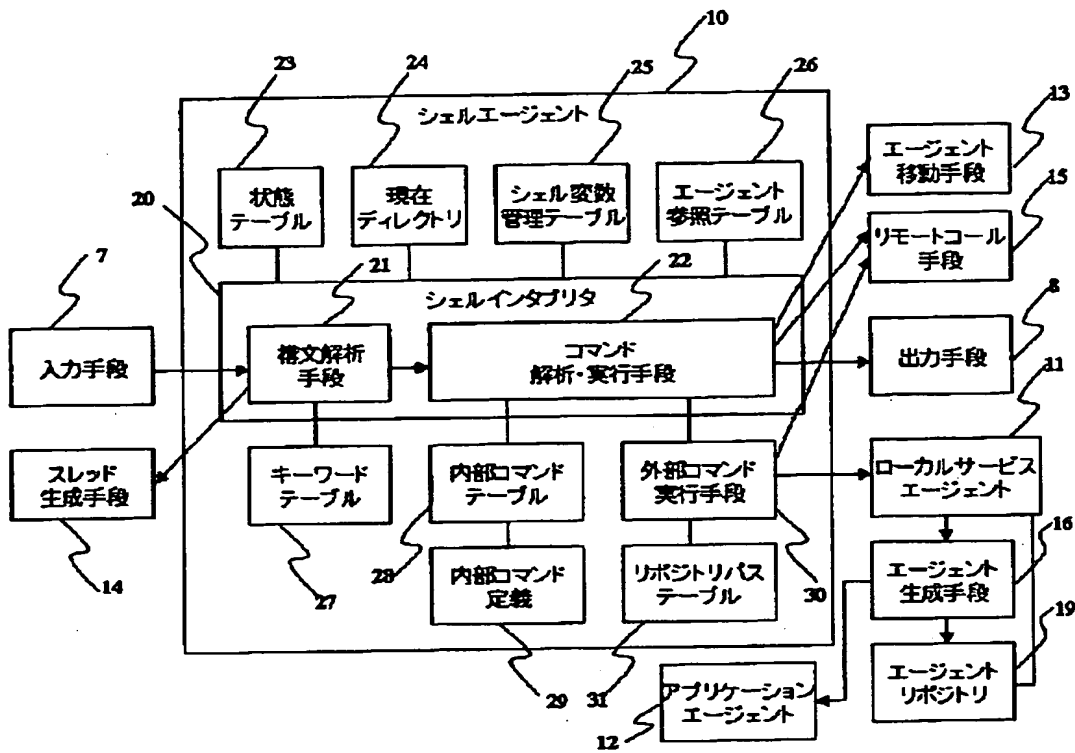




【図 3】



【図 4】

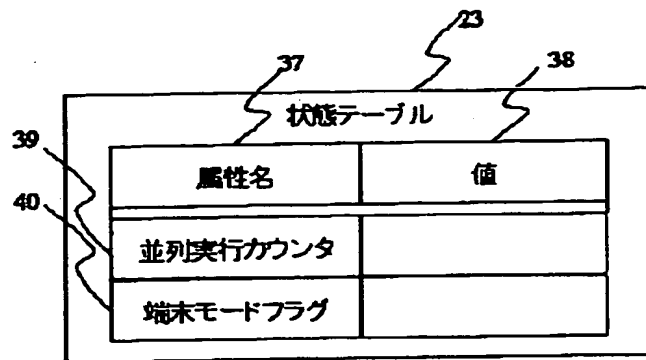


【図 5】

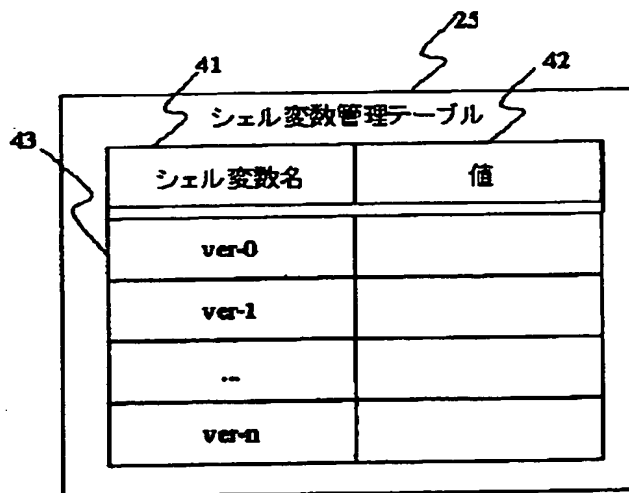
Figure 5 shows a table structure for managing attributes. The table has two columns: **属性名** (Attribute Name) and **値** (Value). The rows represent different types of attributes or bases.

属性名	値
ルートベース	
現在ベース	
ベース内ディレクトリ	

【図 6】



【図 7】

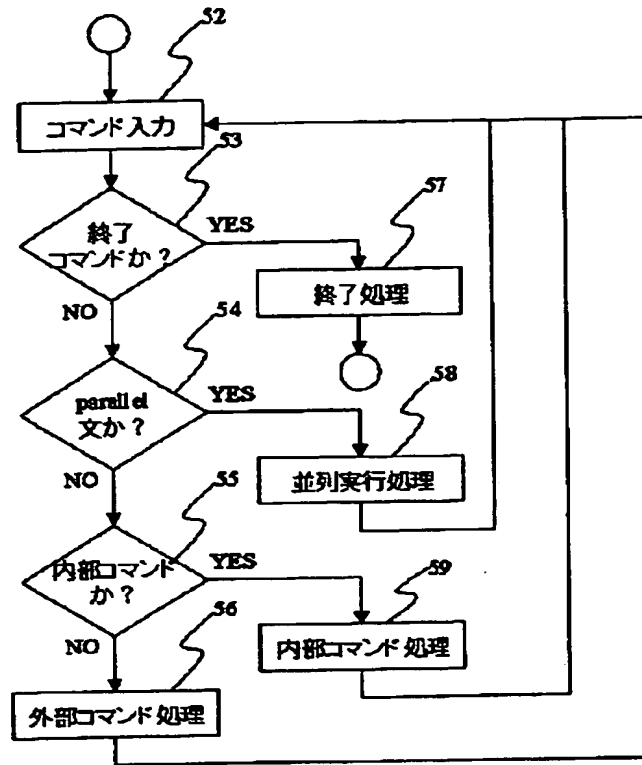


【図 8】

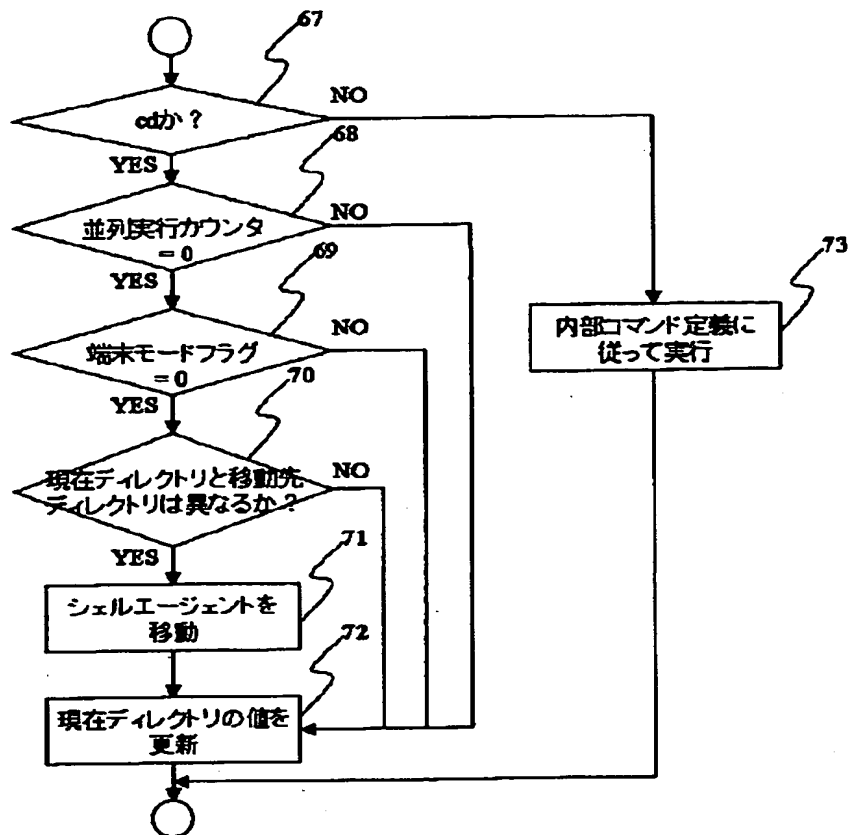
The diagram shows a rectangular box representing the 'Agent Reference Table' (エージェント参照テーブル). A callout line labeled '26' points to the top of the box. Inside the box, at the top, is the label 'エージェント参照テーブル' with a callout line labeled '44'. Below this label is a table with two columns. The first column is labeled 'エージェント変数名' (Agent Variable Name) and the second column is labeled '外部エージェント' (External Agent). The table contains four rows of data: 'agent-0', 'agent-1', '..', and 'agent-n'. A callout line labeled '45' points to the right side of the table, and a callout line labeled '46' points to the left side of the table.

エージェント変数名	外部エージェント
agent-0	
agent-1	
..	
agent-n	

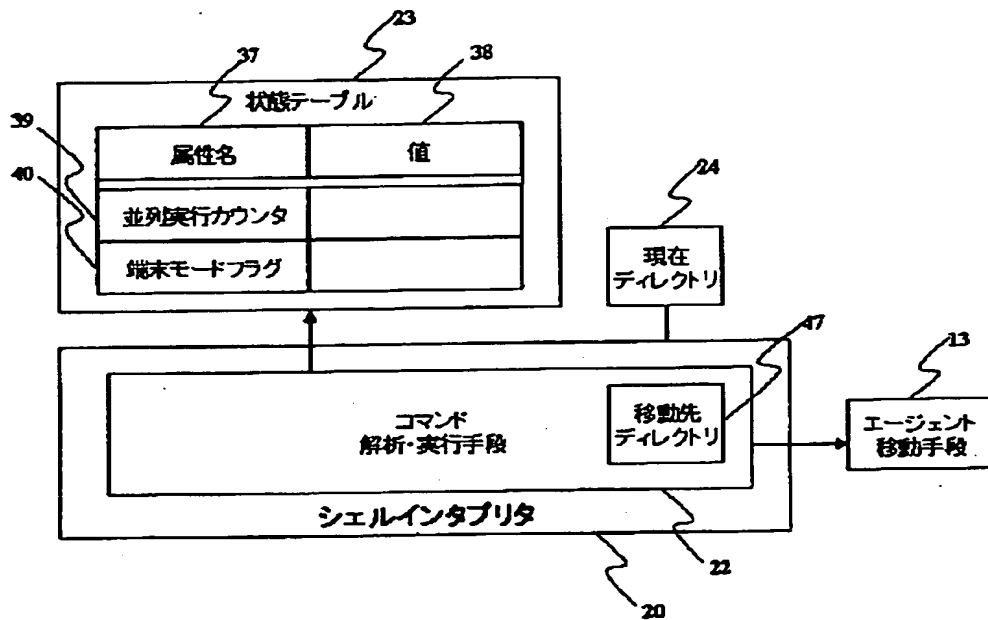
【図 9】



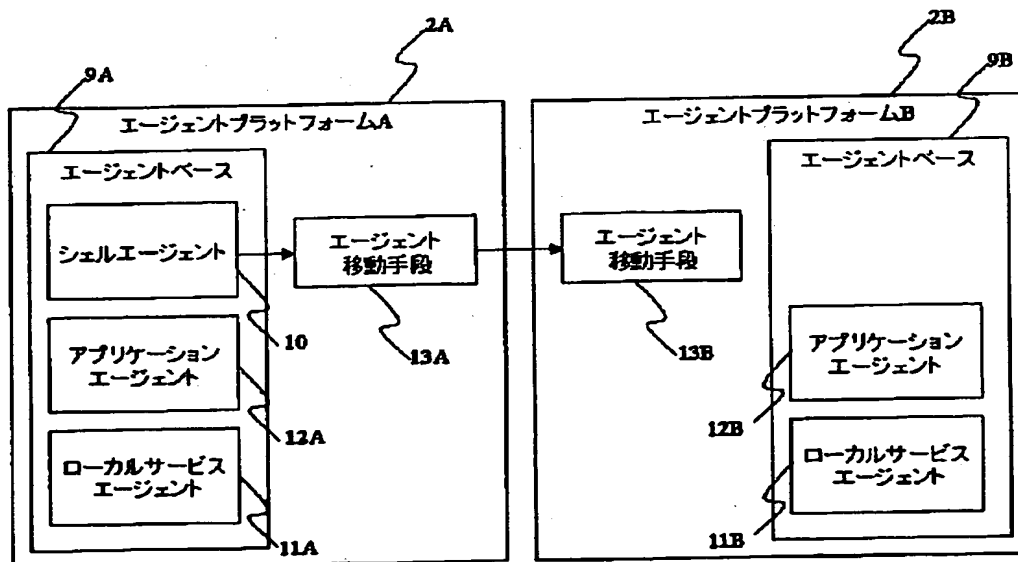
【図 10】



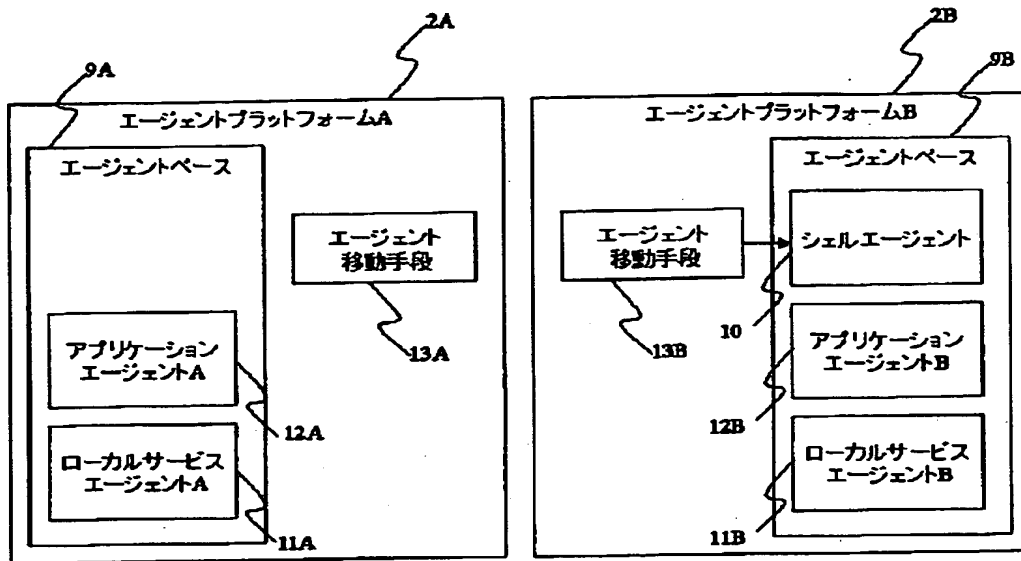
【図 11】



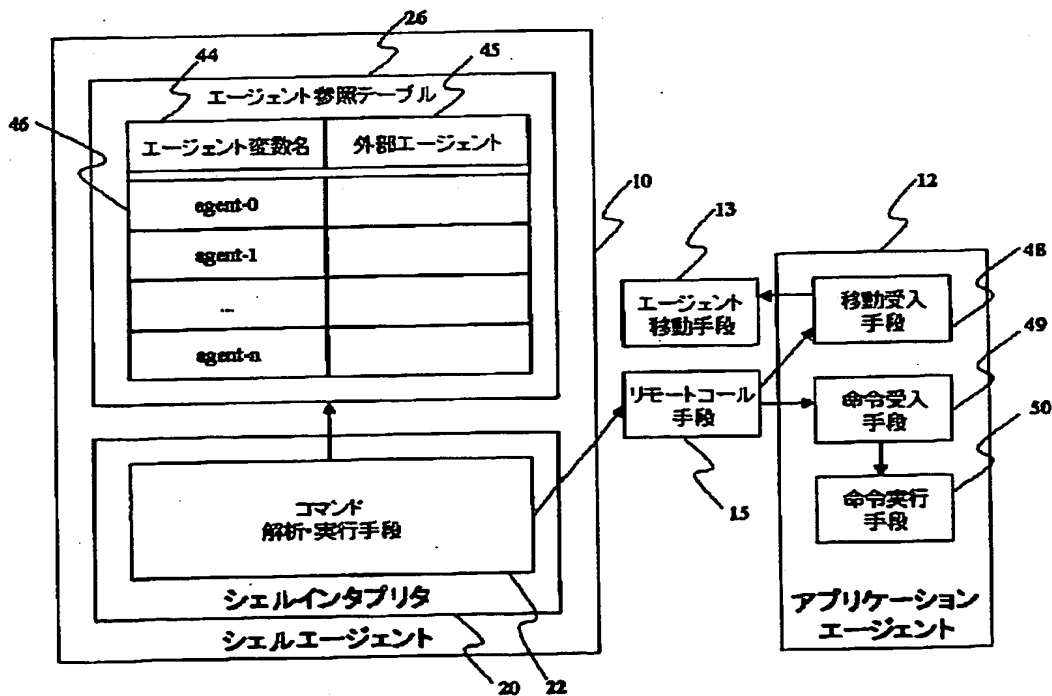
【図 12】



【図 13】

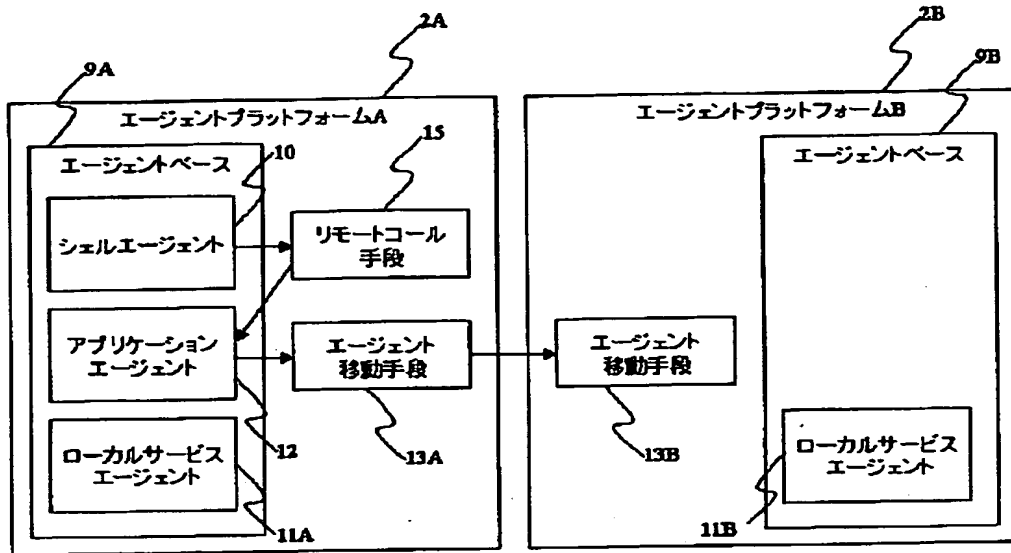


【図 14】

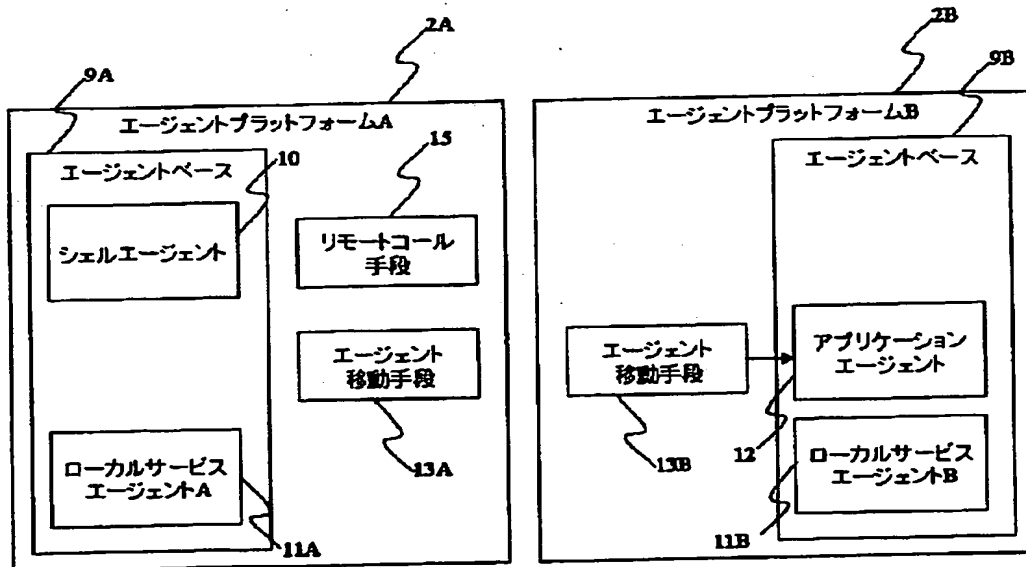




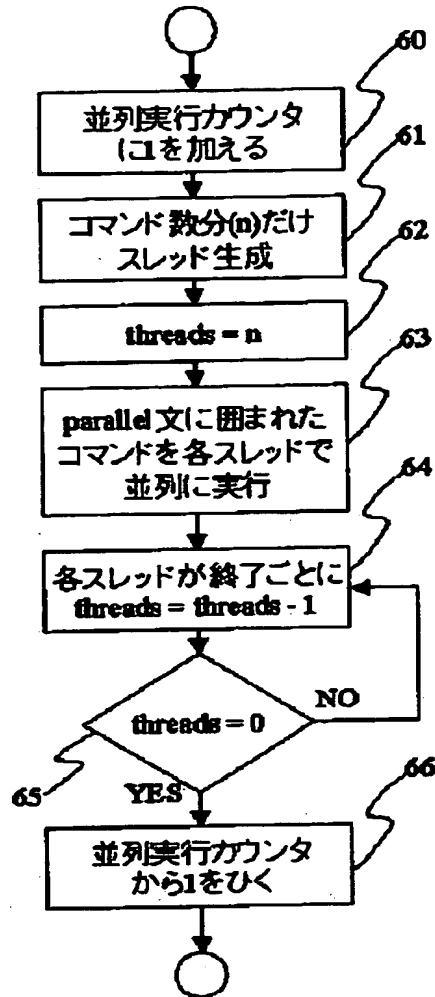
【図 15】



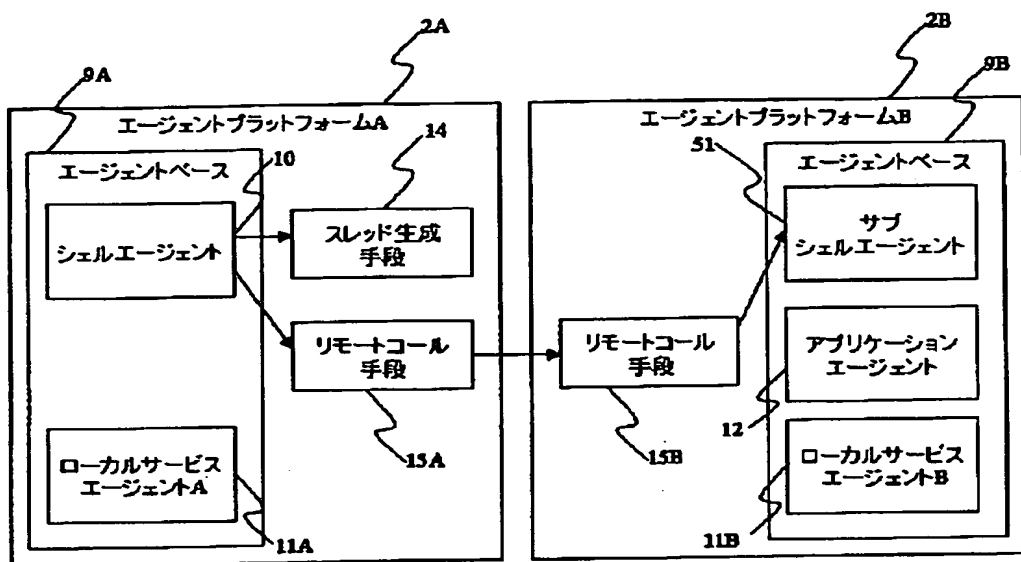
【図 16】



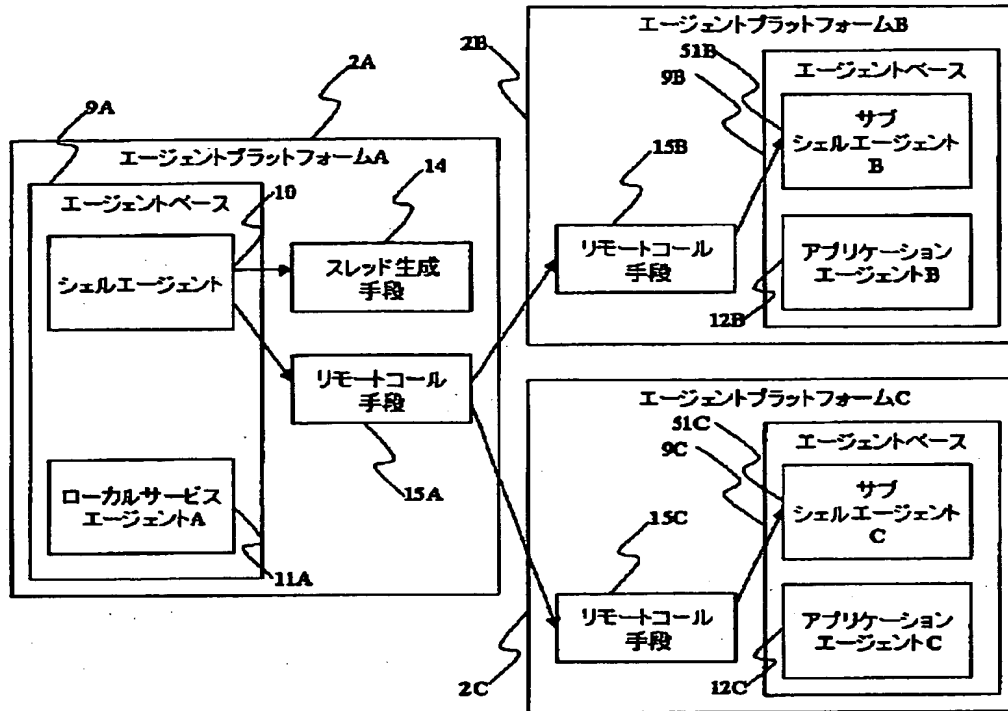
【図 17】



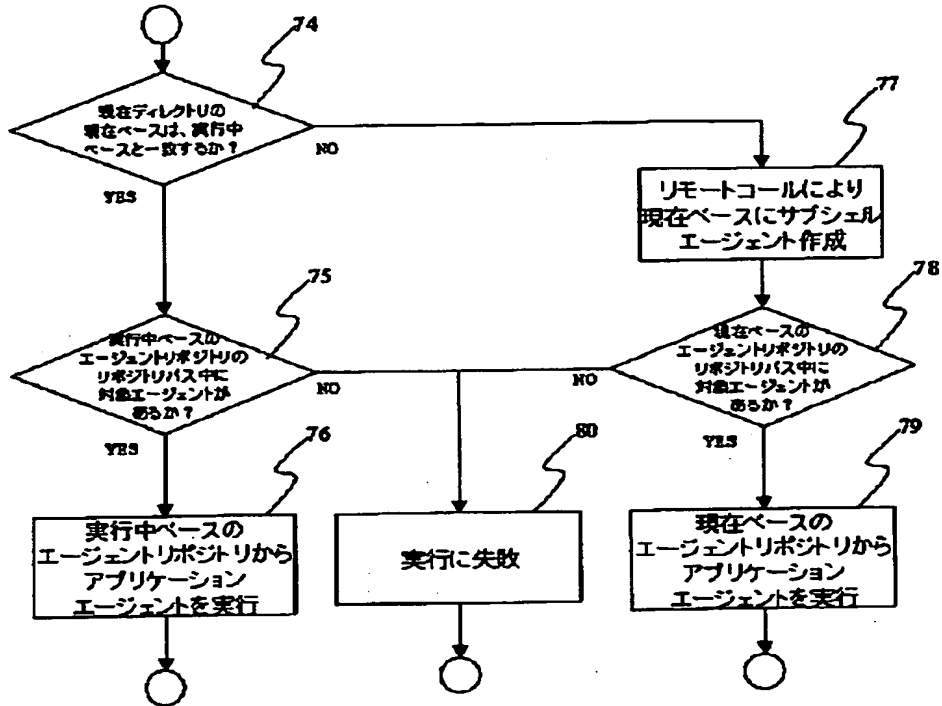
【图 18】



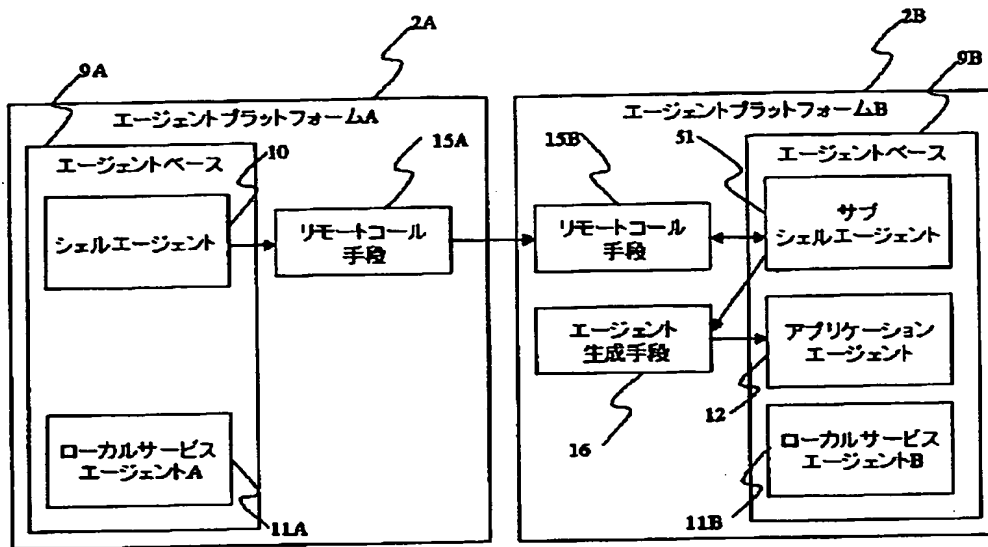
【図 19】



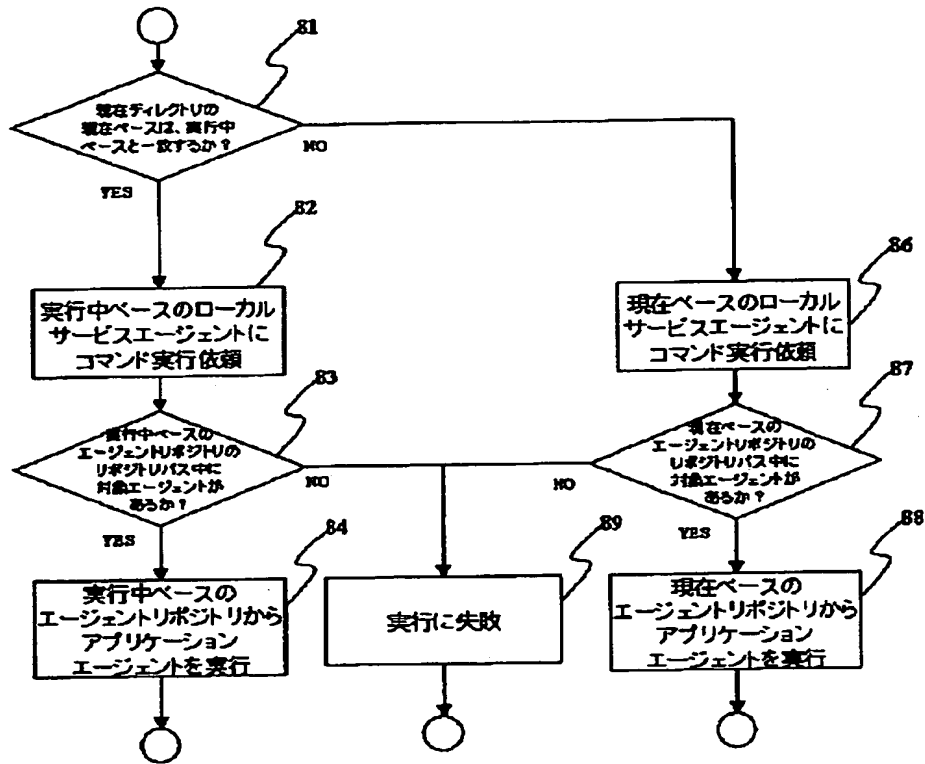
【図 2 0】



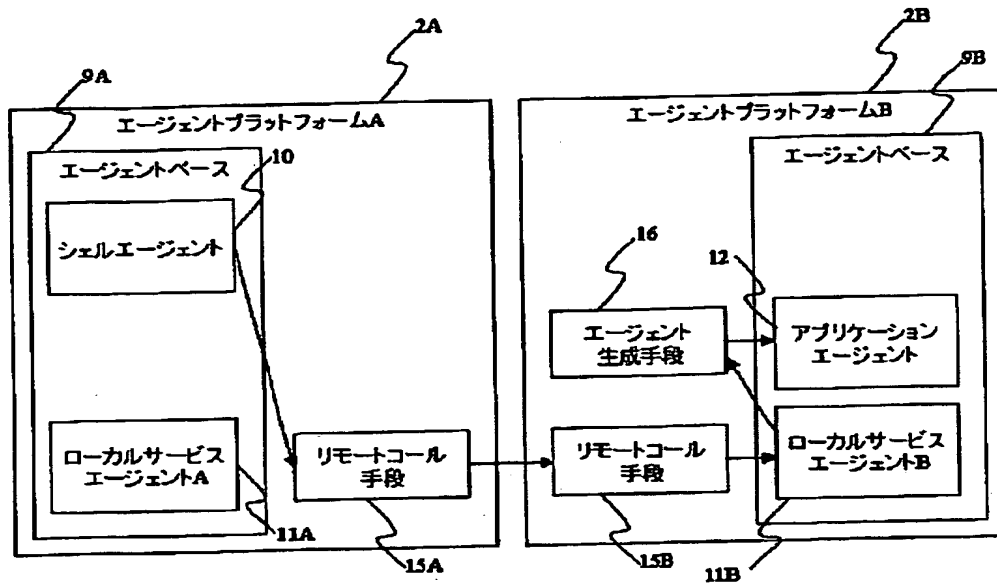
【図 2 1】



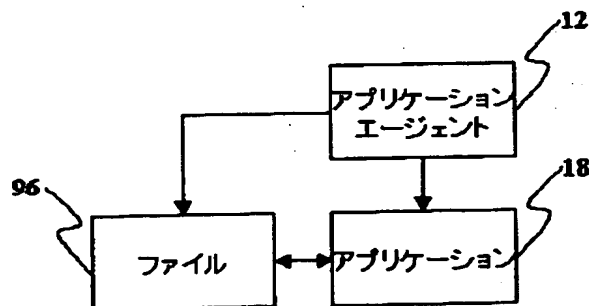
【図 2 2】



【図 2 3】

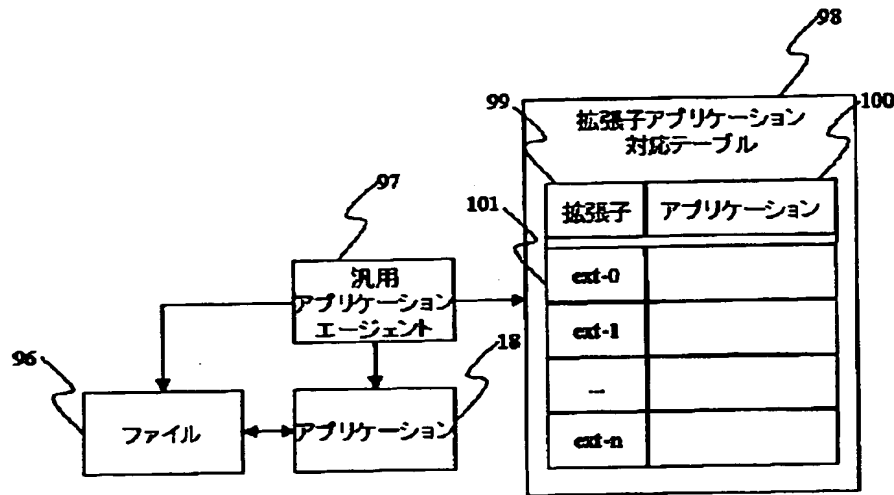


【図 2 4】

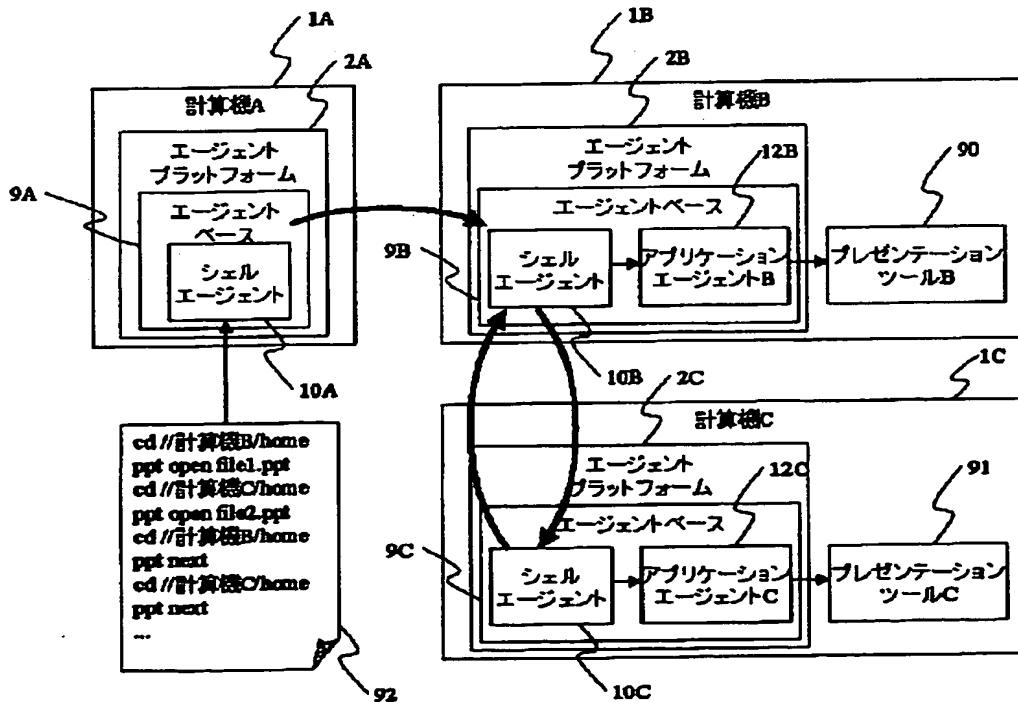




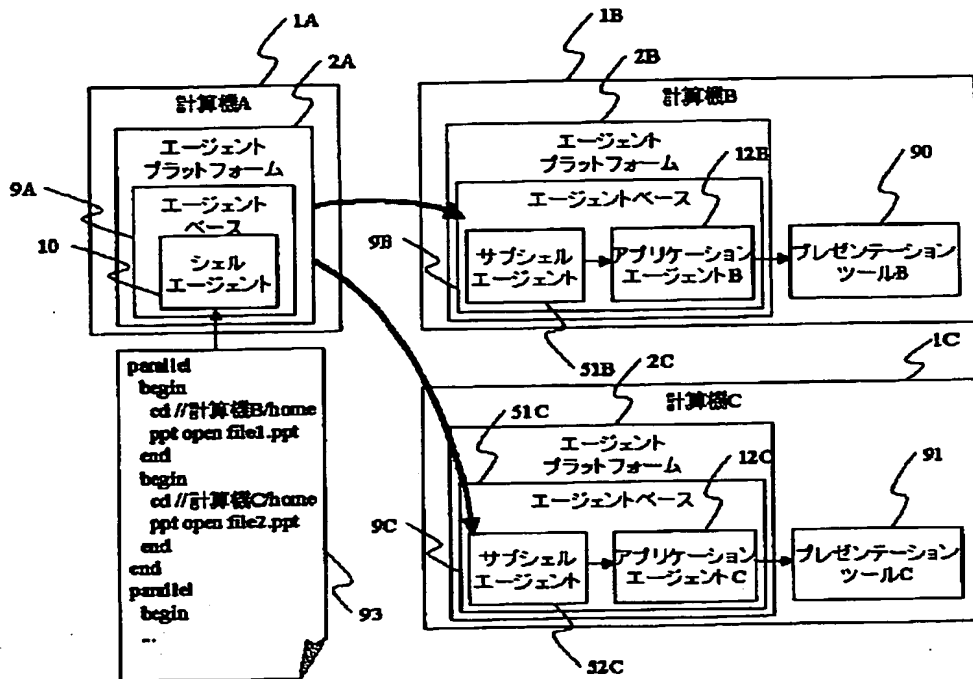
【図 25】



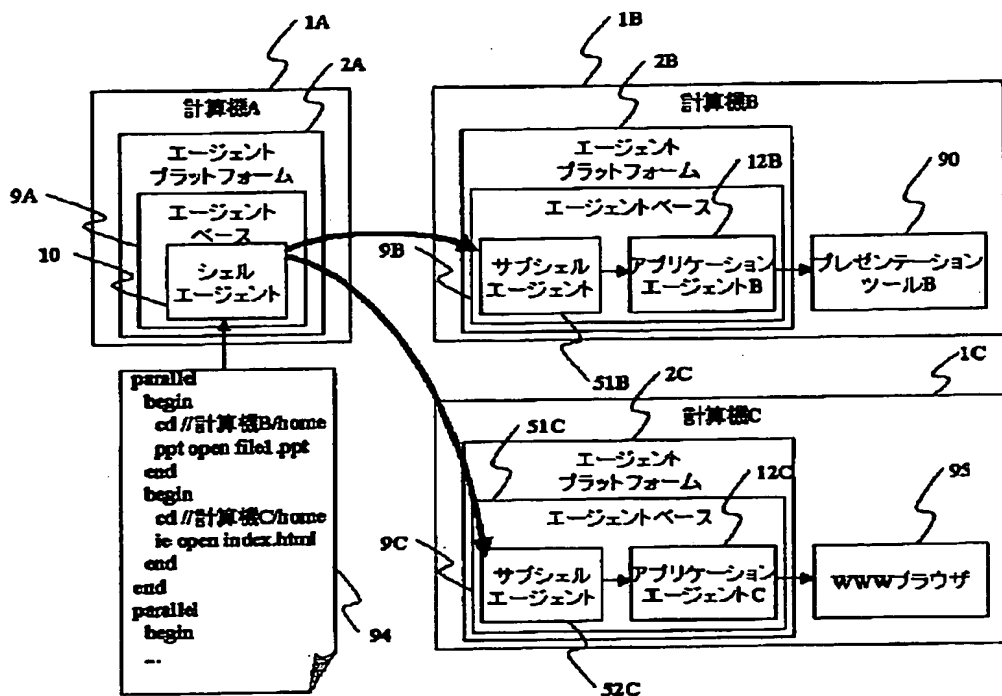
【図 26】



【図 2 7】



【図 28】



【書類名】 要約書

【要約】

【課題】

複数の計算機に分散されたアプリケーションの実行を、簡単なスクリプト言語を用いて制御し、複数のアプリケーションを連携させることで上位の分散アプリケーションを容易に構築可能とするシステムの提供。

【解決手段】

各計算機は、エージェントプラットフォーム 2 に、シェルエージェント 1 0 とローカルサービスエージェント 1 1 とアプリケーションエージェント 1 2 を持つことのできるエージェントベース 9 と、エージェント移動手段 1 3、スレッド生成手段 1 4、リモートコール手段 1 5、エージェント生成手段 1 6 を備え、入力手段 7 を通して入力されたスクリプト言語をシェルエージェント 1 0 が解釈しアプリケーションエージェント 1 2 を起動し、アプリケーションエージェントが実際のアプリケーション 1 8 を管理し、シェルエージェント、アプリケーションエージェントはエージェント移動手段 1 3 を用いて別の計算機に移動でき、リモートコール手段 1 5 を用いて別の計算機中のエージェントと通信できる。

【選択図】

図 2

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社